

EMNLP 2026

**The Seventh Workshop on Insights from Negative Results in
NLP**

Proceedings of the Workshop

October 29, 2026

The EMNLP organizers gratefully acknowledge the support from the following sponsors.

Silver



©2026 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-480-4

Introduction

Publication of negative results is difficult in most fields, and the current focus on benchmark-driven performance improvement exacerbates this situation and implicitly discourages hypothesis-driven research. As a result, the development of NLP models often devolves into a product of tinkering and tweaking, rather than science. Furthermore, it increases the time, effort, and carbon emissions spent on developing and tuning models, as the researchers have little opportunity to learn from what has already been tried and failed.

The mission of the workshop on Insights from Negative Results in NLP is to provide a venue for many kinds of negative results, with the hope that they could yield useful insights and provide a much-needed reality check on the successes of deep learning models in NLP. In particular, we solicit the following types of contributions:

- broadly applicable recommendations for training/fine-tuning, especially if X that didn't work is something that many practitioners would think reasonable to try, and if the demonstration of X's failure is accompanied by some explanation/hypothesis;
- ablation studies of components in previously proposed models, showing that their contributions are different from what was initially reported;
- datasets or probing tasks showing that previous approaches do not generalize to other domains or language phenomena;
- trivial baselines that work suspiciously well for a given task/dataset;
- cross-lingual studies showing that a technique X is only successful for a certain language or language family;
- experiments on (in)stability of the previously published results due to hardware, random initializations, preprocessing pipeline components, etc;
- theoretical arguments and/or proofs for why X should not be expected to work;
- demonstration of issues with under-reporting of training details of pre-trained models, including test data contamination and invalid comparisons.

The seventh iteration of the Workshop on Insights from Negative Results attracted 37 submissions. We accepted 19 papers of these, resulting in a 51% acceptance rate. We hope the workshop will continue to contribute to the many reality-check discussions on progress in NLP. If we do not talk about things that do not work, it is harder to see what the biggest problems are and where the community effort is the most needed.

Organizing Committee

Chairs

João Sedoc, New York University, USA
Aleksandr Drozd, AI-KIP, Japan
Arjun Reddy Akula, Google DeepMind, USA
Raphael Shu, Acenta AI, USA
Anna Rogers, ITU Copenhagen, Denmark
Xiaoli Shen, Microsoft, Japan
Salvatore Giorgi, Vanderbilt University, USA
Adrian Benton, Google, USA

Program Committee

Chairs

João Sedoc, New York University, USA
Aleksandr Drozd, AI-KIP, Japan
Arjun Akula, Google DeepMind, USA
Raphael Shu, Acenta AI, USA
Anna Rogers, ITU Copenhagen, Denmark
Xiaoli Shen, Microsoft, Japan
Salvatore Giorgi, Vanderbilt University, USA
Adrian Benton, Google Research, USA

Program Committee

Sharada Acharya, Khushan Adatiya, Wazir Ali, Abhinav Arora

Stefanie Anna Baby, Susanna Maria Baby, Sanket Badhe, Nihal Balani, João Batista, Uri Berger

Suyash Damle, Aleksandr Drozd

Tamás Ficsor

Adithya V Ganesan, Sheela Ramprakash Goenka, Santhosh Guntupalli, Bhaskar Gurram

Vanya Jauhal

Lakshmojee Koduru, Gaurav Kumar, Laida Kushnareva

Seolhwa Lee, Yifei Li, Zilinghan Li

Afshaan Mazagonwalla, Vilobh Meshram, Kunal Mulwani

Rahul Nagurtha, Juan Navarro-Horniacek

John E. Ortega

Rohit Patwa, Sergey Pletenev, Lahari Poddar, Giovanni Puccetti

Sunny Rai, Tejasvi Ravi

Mario Sanz-Guerrero, Vivek Sardeshmukh, Stephanie Schoch, Erica K. Shimomoto, Mantek Singh, Shashanka Subrahmanya

Emil Vatai, Shubham Vatsal, Shreenivas Venkataramanan, Supriya Vijay

Yichen Wang

Letian Xu

Keynote Talk

Anya Belz

Dublin City University

Keynote Talk

Ryan Boyd

Vanderbilt University

Keynote Talk

Dirk Hovy

Bocconi University

Table of Contents

<i>Geometric Separation Without Functional Specialization: Auditing Subspace Disentanglement in Dual-Encoder LLM Paraphrase Detection</i>	
Govind Arun, Ashish Abraham and Gadha Lekshmi P	1
<i>Exploring Dowker Homology for Sentence Similarity</i>	
Marius Huber and Juri Opitz	7
<i>Does Pragmatic Transfer Help Euphemism Detection? Evidence from Target-Span Marking</i>	
Whitney Poh, Noriko Takahashi, Patrick Lee, Julia Sammartino, Libby Barak, Jing Peng and Anna Feldman	19
<i>Risk-Averse Failure Modes of Jensen-Shannon Distance in RLVR Distributional Alignment</i>	
Sebastian Loftus, Xinpeng Wang, Marie-Catherine De Marneffe and Barbara Plank	33
<i>Controlled Fine-Tuning for Two-Tower Recommenders: Why Gradual Unfreezing Can Fail and Cosine Regularization Suffices</i>	
Gabriel Dershowitz and Djallel Bouneffouf	43
<i>Do Soft Prompts Have Useful Wavelet Structure? Negative Evidence In A Controlled Setup</i>	
Soumyadeep Dey, Srikanth Tenneti, Bamdev Mishra, Shobhit Niranjan and Mayur Datar	48
<i>Decomposing LLM-Judge–Human Agreement: How Much Is Inherited Surface Preference?</i>	
Gaurav Kumar	60
<i>The Flip-Test: A Behavioral Diagnostic for Multi-Agent Debate Failures</i>	
Julia Hu, Alfred Shen and Kumar Lakshminpathi	70
<i>Activation-Based Active Learning for In-Context Learning: Challenges and Insights</i>	
Yaseen Osman, Geoff Merrett and Stuart Middleton	76
<i>The Architecture Was Doing the Work: Why NLI Debiasing Results Do Not Generalise</i>	
Prabhjot Singh	84
<i>When Learned Context Planning Fails to Beat Strong Retrieval: A Controlled Study of Planning, Routing, and Reranking for Long-Context QA</i>	
Yingrui Li and Han Chen	90
<i>No Optimal Language Set Exists for Multilingual Instruction Tuning: Insights from a Linguistically-Informed Study</i>	
Gürkan Soykan and Gözde Şahin	95
<i>Short-Context Gains, Long-Context Drift: Negative Results for Token-Conditioned RoPE</i>	
Allan Kazakov	102
<i>When Gradient Importance Lies: Adaptive LoRA Rank Allocation Fails Under GRPO</i>	
Yash Sawant	110
<i>Do Not Score the Edited Answer: A Negative Result on Loose Evaluation for Verifiable Instruction Following</i>	
José Luis Delgado	115
<i>Perturbation, Not Provenance: Why Activation Steering Does Not Make a Robust Text Watermark</i>	
Sergey Pletenev	120

<i>PEFT Gains Need Hidden Baselines: Rank and Optimizer Sweeps Change LoRA Conclusions</i>	
Ding Wu	127
<i>When Relation-Free Retrieval Is Not Relation-Aware: A Directionality Stress Test for Linear Graph RAG</i>	
Ding Wu	133

Program

Thursday, October 29, 2026

09:05 - 09:50 *Invited Talk 1: Anya Belz*

09:50 - 10:30 *Oral Session 1: Semantic Representation*

Geometric Separation Without Functional Specialization: Auditing Subspace Disentanglement in Dual-Encoder LLM Paraphrase Detection

Govind Arun, Ashish Abraham and Gadha Lekshmi P

Exploring Dowker Homology for Sentence Similarity

Marius Huber and Juri Opitz

Does Pragmatic Transfer Help Euphemism Detection? Evidence from Target-Span Marking

Whitney Poh, Noriko Takahashi, Patrick Lee, Julia Sammartino, Libby Barak, Jing Peng and Anna Feldman

11:00 - 11:45 *Invited Talk 2: Ryan Boyd*

11:45 - 12:25 *Oral Session 2: Tuning Recipes that Fail*

Risk-Averse Failure Modes of Jensen-Shannon Distance in RLVR Distributional Alignment

Sebastian Loftus, Xinpeng Wang, Marie-Catherine De Marneffe and Barbara Plank

Controlled Fine-Tuning for Two-Tower Recommenders: Why Gradual Unfreezing Can Fail and Cosine Regularization Suffices

Gabriel Dershowitz and Djallel Bouneffouf

Do Soft Prompts Have Useful Wavelet Structure? Negative Evidence In A Controlled Setup

Soumyadeep Dey, Srikanth Tenneti, Bamdev Mishra, Shobhit Niranjan and Mayur Datar

14:05 - 14:50 *Invited Talk 3: Dirk Hovy*

14:50 - 15:30 *Oral Session 3: Test-Time Behavior and Evaluation*

Decomposing LLM-Judge–Human Agreement: How Much Is Inherited Surface Preference?

Gaurav Kumar

Thursday, October 29, 2026 (continued)

The Flip-Test: A Behavioral Diagnostic for Multi-Agent Debate Failures

Julia Hu, Alfred Shen and Kumar Lakshmipathi

Activation-Based Active Learning for In-Context Learning: Challenges and Insights

Yaseen Osman, Geoff Merrett and Stuart Middleton

16:05 - 17:30

Poster Session

The Architecture Was Doing the Work: Why NLI Debiasing Results Do Not Generalise

Prabhjot Singh

When Learned Context Planning Fails to Beat Strong Retrieval: A Controlled Study of Planning, Routing, and Reranking for Long-Context QA

Yingrui Li and Han Chen

No Optimal Language Set Exists for Multilingual Instruction Tuning: Insights from a Linguistically-Informed Study

Gürkan Soykan and Gözde Şahin

Short-Context Gains, Long-Context Drift: Negative Results for Token-Conditioned RoPE

Allan Kazakov

When Gradient Importance Lies: Adaptive LoRA Rank Allocation Fails Under GRPO

Yash Sawant

Do Not Score the Edited Answer: A Negative Result on Loose Evaluation for Verifiable Instruction Following

José Luis Delgado

Perturbation, Not Provenance: Why Activation Steering Does Not Make a Robust Text Watermark

Sergey Pletenev

PEFT Gains Need Hidden Baselines: Rank and Optimizer Sweeps Change LoRA Conclusions

Ding Wu

Thursday, October 29, 2026 (continued)

*When Relation-Free Retrieval Is Not Relation-Aware: A Directionality Stress Test
for Linear Graph RAG*

Ding Wu

Geometric Separation Without Functional Specialization: Auditing Subspace Disentanglement in Dual-Encoder LLM Paraphrase Detection

Govind Arun

University of Maryland, College Park
College Park, MD, USA

Ashish Anton Abraham*

Indira Gandhi National Open University
New Delhi, India

Gadha Lekshmi P

Rutgers University, New Brunswick
New Brunswick, NJ, USA

Abstract

Explicit subspace disentanglement, enforcing orthogonality and simplex constraints on learned projections, is an increasingly common recipe for paraphrase models claimed to separate lexical, syntactic, and semantic information. A dual-encoder hierarchical model trained with these constraints on the DiPPER corpus of LLM-generated paraphrases produces near-perfect geometric separation, a pairwise Centered Kernel Alignment (CKA) of roughly 0.0007, yet that separation is not evidence of functional linguistic specialization. Three results establish this. First, near-zero CKA is largely preordained by an explicit orthogonality penalty, which converges to 4×10^{-6} . Second, each of the three subspaces, including the nominally syntactic one for which we identify no functional probe, classifies paraphrase in isolation at F_1 above 0.986, so the subspaces are not functionally distinct. Third, removing disentanglement at matched capacity costs only 0.013 F_1 , leaving most of the reported gain attributable to dual-encoder capacity rather than disentanglement. Capacity-matched ablations and controlled functional probes, not CKA alone, are required to substantiate disentanglement claims; orthogonality penalties otherwise manufacture the very metric used to certify them.

1 Introduction

A recurring design pattern in representation learning for paraphrase and semantic-matching tasks splits an encoder output into several projected subspaces and pushes those subspaces apart with explicit penalties, typically a soft orthogonality term on the projection matrices and a simplex-style term that asks the projections to jointly preserve the original representation. The subspaces are then labeled lexical, syntactic, and semantic, and low pairwise representational similarity is offered as

evidence of linguistically specialized features. Reporting a near-zero similarity metric, most often CKA (Kornblith et al., 2019), as confirmation that disentanglement worked has become routine.

The pattern rests on real evidence: probing studies show that transformer encoders distribute lexical, syntactic, and semantic information across layers and subspaces rather than token positions (Hewitt and Manning, 2019; Tenney et al., 2019), and several paraphrase systems decompose representations along these lines or combine encoders for complementary signal (Xu et al., 2020; Liu et al., 2018; Ortiz-Barajas et al., 2022; Peng et al., 2022; Yang et al., 2016). Driving inter-subspace similarity to near zero therefore appears to operationalize that picture.

This appearance does not survive scrutiny. The instance audited here is a dual-encoder hierarchical transformer (DeBERTa-v3 and RoBERTa) that projects each encoder output into three subspaces, applies orthogonality and simplex penalties, attaches level-specific auxiliary supervision under standard multi-task weighting (Dan et al., 2022; He et al., 2022), and fuses the levels through a gated bilinear meta-classifier. Trained on the DiPPER corpus (Krishna et al., 2023), it reaches $F_1 = 0.9846$ with almost perfectly separated subspaces under CKA. Three results recomputed on the trained model overturn that reading: near-zero CKA is mechanically induced by the orthogonality penalty; every subspace classifies the task in isolation at F_1 above 0.986; and removing disentanglement at matched capacity costs only 0.013 F_1 , leaving the gain dominated by dual-encoder capacity, that is, a trained ensemble. CKA-as-disentanglement-evidence can therefore mislead, and concrete safeguards follow.¹

* Corresponding author: ashishabraham22@gmail.com

¹Code: <https://github.com/GIND123/Hierarchical-Transformer>.

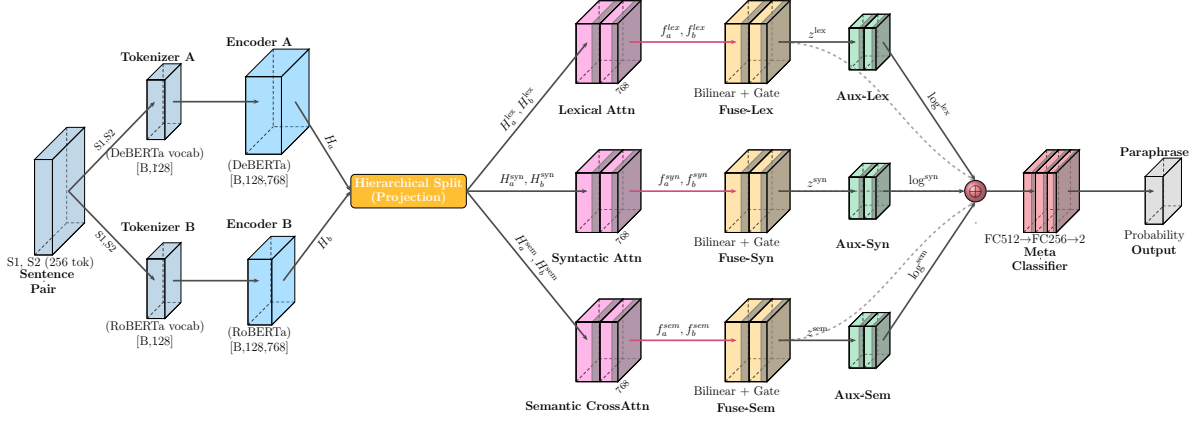


Figure 1: Hierarchical dual-encoder architecture under test: two encoders, level-wise projection into lexical, syntactic, and semantic subspaces, per-level attention and gated bilinear fusion, and a meta-classifier over the fused vectors and auxiliary logits.

2 Method under test

The audited model appears in Figure 1. Two pretrained encoders (DeBERTa-v3-base and RoBERTa-base) produce contextual states $H_a, H_b \in \mathbb{R}^{L \times d}$ for an ordered text pair. Three shared projection matrices $S_\ell \in \mathbb{R}^{d_\ell \times d}$ map each encoder output into level-specific subspaces $\ell \in \{\text{lex}, \text{syn}, \text{sem}\}$:

$$H_a^\ell = H_a S_\ell^\top, \quad H_b^\ell = H_b S_\ell^\top. \quad (1)$$

A soft orthogonality penalty $\mathcal{L}_{\text{ortho}}$ minimizes normalized Frobenius inner products $\langle S_\ell, S_{\ell'} \rangle_F$ for $\ell \neq \ell'$, and a simplex penalty $\mathcal{L}_{\text{simp}}$ constrains $\sum_\ell S_\ell S_\ell^\top \approx I_d$ (Figure 2). Additive attention pools each stream independently:

$$e_i^{(X)} = w_X^\top \tanh(W_X h_i^{(X)}), \quad (2)$$

$$\alpha_i^{(X)} = \frac{\exp(e_i^{(X)})}{\sum_{j=1}^L \exp(e_j^{(X)})}, \quad f_a^X = \sum_{i=1}^L \alpha_i^{(X)} h_i^{(X,a)}. \quad (3)$$

The semantic level adds bidirectional multi-head attention (MHA) with layer normalization (LN), $\tilde{H}_a^{\text{sem}} = \text{LN}(H_a^{\text{sem}} + \text{MHA}(H_a^{\text{sem}}, H_b^{\text{sem}}, H_b^{\text{sem}}))$ and symmetrically for b . Per level, encoder B is aligned to A and the streams are combined by a gated bilinear fusion with a residual gate:

$$b^\ell = \text{Bilinear}(f_a^\ell, W_{\text{align}} f_b^\ell), \quad (4)$$

$$g^\ell = \sigma(W_g[f_a^\ell, W_{\text{align}} f_b^\ell]), \quad (5)$$

$$z^\ell = W_p(g^\ell \odot b^\ell + (1 - g^\ell) \odot f_a^\ell). \quad (6)$$

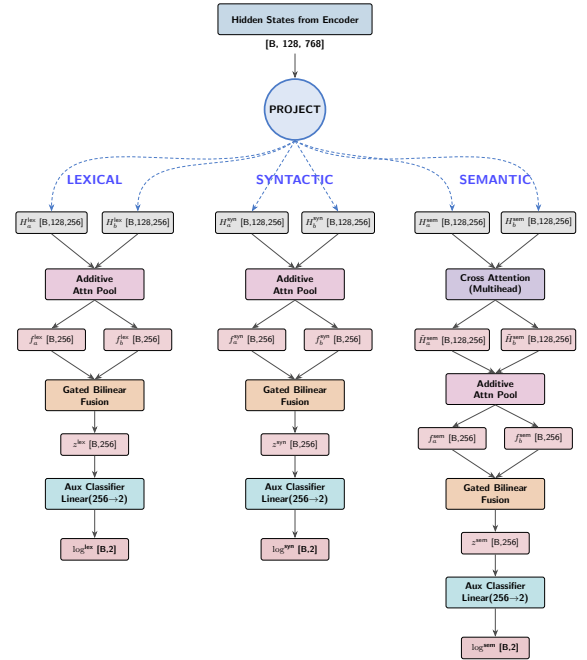


Figure 2: Level-wise splitting of features and the cross-attention applied at the semantic level.

Each level emits an auxiliary logit u^ℓ through a lightweight classifier, and a multi-layer perceptron (MLP) meta-classifier (Figure 3) consumes the three fused vectors and three auxiliary logits, $\hat{y} = \text{MLP}([z^{\text{lex}}; z^{\text{syn}}; z^{\text{sem}}; u^{\text{lex}}; u^{\text{syn}}; u^{\text{sem}}])$. Training minimizes a composite loss

$$\mathcal{L} = \alpha \mathcal{L}_{\text{main}} + (1 - \alpha) \frac{\mathcal{L}_{\text{lex}} + \mathcal{L}_{\text{syn}} + \mathcal{L}_{\text{sem}}}{3} + \lambda_{\text{ortho}} \mathcal{L}_{\text{ortho}} + \lambda_{\text{simp}} \mathcal{L}_{\text{simp}}, \quad (7)$$

with $\alpha = 0.6$, $\lambda_{\text{ortho}} = 0.02$, $\lambda_{\text{simp}} = 0.01$. The level-wise design is motivated by evidence of over-

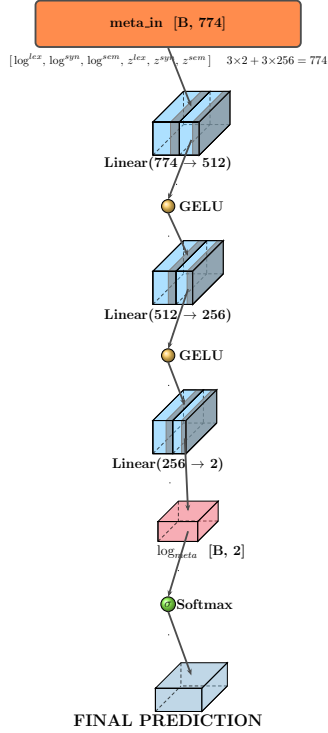


Figure 3: Meta-classifier over the concatenated fused vectors and auxiliary logits.

reliance on superficial lexical cues (Macbeth et al., 2020). What is at issue is not the architecture but the claim attached to it: that the penalties yield interpretable, linguistically specialized subspaces.

3 Setup and probing protocol

Training and evaluation use the DiPPER corpus (Krishna et al., 2023): 1,150,000 training pairs, 50,000 validation pairs, and the official test split of 324,633 pairs, with the decision threshold tuned on validation to $\tau^* = 0.55$. The probing protocol, under-reported in the original study, is specified as follows. **Pooling**: each subspace projection is pooled independently for each sentence using additive attention pooling. Both sentence-level pooled vectors are combined via a gated bilinear fusion to yield the pair representation used for probing. **CKA**: linear CKA (Kornblith et al., 2019) between the pooled projected representations of the three subspaces over the validation set, the metric the orthogonality penalty most directly influences. **Linear probing**: for each subspace independently, the pooled pair representation is used to fit an L2-regularized logistic regression on the binary paraphrase label with an L2 penalty strength of 1.0, and validation F_1 reported; a subspace that classifies

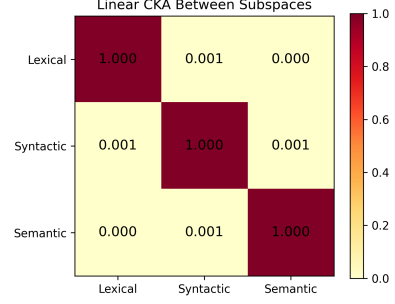


Figure 4: CKA between the learned subspaces. The near-zero off-diagonal values are driven by the orthogonality penalty ($\mathcal{L}_{\text{ortho}} = 4 \times 10^{-6}$) and restate constraint satisfaction rather than independent evidence of linguistic specialization.

well in isolation solves the task on its own. No gradients propagate into the encoder or projection layers during probing. **Probe data and splits**: each probe is trained on the first 20,000 validation pairs, split into 16,000 training and 4,000 held-out examples; this split is drawn from the head of the validation file rather than an independently reshuffled sample, so the class balance of the probing subset is not separately verified against the overall validation positive rate. **Functional grounding**: Spearman correlation between the L2 norm of a subspace representation and surface statistics, namely token Jaccard overlap and length ratio for the lexical claim and surface lexical overlap for the semantic claim. No surface-free syntactic probe was identified here, leaving standard structural probes to future work; this absence is treated below as part of the result.

4 Findings

4.1 Geometric separation is not functional disentanglement

The subspaces are almost perfectly separated: CKA is 0.0008 (lexical, syntactic), 0.0005 (lexical, semantic), and 0.0007 (syntactic, semantic), with $\mathcal{L}_{\text{ortho}}$ converging to 4×10^{-6} (Figure 4). The original study read these values as confirmation that the constraints disentangle the linguistic representation spaces. That inference is close to circular. The orthogonality penalty minimizes inter-projection alignment, which in turn drives pairwise CKA toward zero, so a near-zero value mostly restates constraint satisfaction rather than showing that the subspaces encode distinct linguistic levels. CKA under an explicit orthogonality objective is not an interpretability metric for this purpose.

Corpus	Pooled CKA			Probe F_1			
	l/s	l/m	s/m	lex	syn	sem	rand
DiPPER	0.001	0.001	0.001	> 0.986			–
PAWS	0.981	0.967	0.951	0.745	0.738	0.733	0.759
QQP	0.992	0.973	0.957	0.756	0.752	0.762	0.760

Table 1: The frozen DiPPER-trained checkpoint evaluated on two out-of-domain corpora, 2,000 evaluation pairs each, probes averaged over five seeds. Subspace pairs are lexical/syntactic (l/s), lexical/semantic (l/m), and syntactic/semantic (s/m). Pooled CKA reverses from near zero to near one while the projection matrices are unchanged. A random projection of the raw encoder states, matched in dimension, exceeds all three named subspaces on PAWS and matches them to within seed variation on QQP, so the subspace labels carry no probe-relevant information. The frozen meta-classifier attains ROC AUC 0.442 on PAWS and 0.713 on QQP, against 0.998 in domain.

Subspace	Surface correlate (ρ)	Probe F_1
Lexical	Jaccard 0.479, length 0.442	> 0.986
Syntactic	none identifiable	> 0.986
Semantic	surface overlap 0.228	> 0.986

Table 2: Functional probing of the three subspaces. Every subspace classifies paraphrase in isolation at $F_1 > 0.986$, yet the grounding is uneven: the lexical subspace has a clear surface correlate, the semantic subspace stays tied to surface overlap, and no functional correlate was identified for the syntactic subspace. Equal task performance with unequal grounding indicates the subspaces are not linguistically specialized.

The failure is sharper than circularity alone. Linear CKA between two projected representations is governed by $\|S_\ell \Sigma S_{\ell'}^\top\|_F$, where Σ is the covariance of the pooled encoder states. The penalty drives $S_\ell S_{\ell'}^\top$ toward zero, but that forces the cross term to vanish only when Σ is close to isotropic on the relevant directions. CKA is therefore a joint property of the projections and the evaluation distribution, not of the model alone. Applying the same checkpoint without retraining to PAWS (Zhang et al., 2019) and QQP (Wang et al., 2018) makes the dependence explicit (Table 1): the projections still satisfy the constraint, with pairwise Frobenius cosines below 10^{-5} , yet pooled CKA rises to 0.95 through 0.99 across all three subspace pairs. One set of weights certifies near-perfect separation on one corpus and near-total similarity on another. A metric that reverses under distribution shift while the constraint it supposedly reflects stays satisfied cannot serve as evidence of learned specialization.

4.2 All subspaces solve the task equally

The decisive evidence is functional. Linear probing on each subspace in isolation yields individual F_1 above 0.986 for all three, comparable to the full model’s 0.9846. If the lexical subspace alone predicts paraphrase as well as the semantic subspace, the subspaces are not functionally distinct on the task, contradicting the framing that each captures a different linguistic granularity. The only clean functional signal is lexical: lexical magnitude correlates with token Jaccard overlap ($\rho = 0.479$) and length ratio ($\rho = 0.442$). The semantic subspace is not cleanly abstract, retaining $\rho = 0.228$ with surface overlap, and we identify no functional probe for the syntactic subspace (Table 2). The juxtaposition is the point: a subspace with no identifiable syntactic signal classifies paraphrase as well as the others. Orthogonality has produced three redundant solvers of the same task, not a lexical, a syntactic, and a semantic specialist.

4.3 The gain is capacity, not disentanglement

The ablation in Table 3 is reread for what it controls. The full model scores $F_1 = 0.9846$. A dual-encoder variant with the same two encoders and training procedure but a single shared projection and no auxiliary losses, “Fusion Without Disentanglement,” scores 0.9715. That variant is near parameter-matched to the full model, differing only in one projection versus three, well under one percent of the 361.5M total. The three-way split plus auxiliary supervision therefore buys 0.013 F_1 . The larger increments over the single encoders (+0.018 over DeBERTa-v3, +0.023 over RoBERTa) track the addition of a second encoder, that is, capacity and fusion in the manner of a trained ensemble. The foregrounded architectural novelty contributes the smallest of the three increments, far below what its complexity implies, at a large cost: 361.5M parameters and 0.52 ms per sample against 124.6M and 0.09 ms for the RoBERTa baseline.

5 Discussion and recommendations

One mechanism unifies the three findings: an explicit penalty shaped the representation and, through a metric sensitive to that same penalty, certified the shaping as meaningful. Low CKA was induced by the objective, equal probe performance shows the resulting subspaces are interchangeable, and the capacity-matched ablation shows the elaboration adds little. The pattern is not confined to

Model	Split	F1	Accuracy	Precision	Recall	AUC ROC	AUC PR	MCC
Full hierarchical model	Test	0.9846	0.9794	0.9825	0.9867	0.9978	0.9989	0.9536
Fusion without disentanglement	Test	0.9715	0.9619	0.9681	0.9750	0.9936	0.9968	0.9140
DeBERTa-v3-base only	Test	0.9661	0.9547	0.9624	0.9699	0.9913	0.9958	0.8977
RoBERTa-base only	Test	0.9619	0.9488	0.9541	0.9699	0.9898	0.9950	0.8841
Majority baseline	Test	0.80	0.67	0.67	1.00	0.50	0.67	0.00

Table 3: Ablation on the DiPPER test split, reread for the capacity control. Disentanglement adds only 0.013 F_1 over a near parameter-matched dual encoder without disentanglement. Following the row order above, $\Delta F_1 = 0.013$, 0.018, and 0.023 against the three learned baselines. The larger increments over the single encoders track the addition of a second encoder, not subspace separation.

representation geometry. Evaluation validity has been shown to differ sharply across environments in multi-agent LLM benchmarks, where failure-heavy settings can reward robustness to opponent errors as much as strategic ability (Wang et al., 2026). In both cases the reported number is sensitive to a factor other than the construct it is taken to measure. Three safeguards follow for work reporting subspace disentanglement. First, a similarity metric directly minimized by the training objective is not evidence of specialization; CKA, cosine, or Frobenius separation between explicitly orthogonalized projections should be reported as constraint satisfaction, not interpretability. Second, a capacity-matched ablation that removes the disentanglement mechanism while holding encoders, data, and training steps fixed is mandatory, since gains from added capacity are otherwise misattributed to the proposed design. Third, each named subspace requires a controlled functional probe, ideally one a non-specialized subspace would fail; a label such as “syntactic” must be earned by a signal that distinguishes it from the other levels, not assumed from its projection.

Limitations

The audit trains a single model on a single corpus; the PAWS and QQP results evaluate that same frozen checkpoint without retraining. For lack of compute, external baselines were not trained on the DiPPER split, so the capacity argument rests on internal ablations and remains DiPPER-only. The probes are task-coupled and small-scale, and a richer battery of surface-free syntactic probes might recover a signal not found here. The conclusion is therefore scoped to this recipe, orthogonality plus simplex penalties with attached level labels, not to all representation disentanglement.

References

- Zhengjia Dan, Yue Zhao, Xiaojun Bi, Licheng Wu, and Qiang Ji. 2022. Multi-task transformer with adaptive cross-entropy loss for multi-dialect speech recognition. *Entropy*, 24(10):1429.
- Yun He, Xue Feng, Cheng Cheng, Geng Ji, Yunsong Guo, and James Caverlee. 2022. Metabalance: Improving multi-task recommendations via adapting gradient magnitudes of auxiliary tasks. In *Proceedings of the ACM Web Conference 2022*, pages 2205–2215.
- John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR.
- Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36:27469–27500.
- Bang Liu, Ting Zhang, Fred X. Han, Di Niu, Kunfeng Lai, and Yu Xu. 2018. Matching natural language sentences with hierarchical sentence factorization. In *Proceedings of the 2018 World Wide Web Conference*, pages 1237–1246.
- J. C. Macbeth, E. Chang, J. G. Chen, and S. Grandic. 2020. A broader range for ‘meaning the same thing’: Human against machine on hard paraphrase detection tasks. In *Proceedings of The Eighth Annual Conference on Advances in Cognitive Systems (ACS-2020)*.
- Jesus-German Ortiz-Barajas, Gemma Bel-Enguix, and Helena Gómez-Adorno. 2022. Sentence-CROBI: A simple cross-bi-encoder-based neural network architecture for paraphrase identification. *Mathematics*, 10(19).

- Qiwei Peng, David Weir, Julie Weeds, and Yekun Chai. 2022. Predicate-argument based bi-encoder for paraphrase identification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5579–5589, Dublin, Ireland. Association for Computational Linguistics.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP*, pages 353–355.
- Kevin Wang, Anna Thöni, Benjamin Kempinski, Bobby Cheng, Jianzhu Yao, Benjamin Finch, Leon Guertler, Viraj Nadkarni, Yihan Jiang, Aliaksei Korshuk, and 1 others. 2026. MINDGAMES: A live arena for evaluating social and strategic reasoning in multi-agent LLMs. *NeurIPS 2025 Datasets and Benchmarks Track*.
- Sheng Xu, Xingfa Shen, Fumiyo Fukumoto, Jiyi Li, Yoshimi Suzuki, and Hiromitsu Nishizaki. 2020. Paraphrase identification with lexical, syntactic and sentential encodings. *Applied Sciences*, 10(12):4144.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.
- Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1298–1308.

Exploring Dowker Homology for Sentence Similarity

Marius Huber Juri Opitz

Department of Computational Linguistics

University of Zürich

{marius.huber, jurialexander.opitz}@uzh.ch

Abstract

Dowker homology is a topological tool that may be used to analyze the relative position of two point clouds living in a common space. We investigate whether Dowker homology captures sentence similarity information by treating the embeddings of the tokens that constitute a sentence pair as a pair of point clouds in the latent space of a transformer model, using both models that have and have not been fine-tuned for sentence similarity. We find that Dowker homology captures sentence similarity information, as measured by regressing Dowker homology features onto ground-truth similarity scores, and that it can be used for visual inspection of similarity data and models. In an attempt to make Dowker homology readily applicable, we derive from it single-number summaries that we expect to capture sentence similarity directly. These turn out to work reasonably well, but without outperforming standard sentence similarity measures based on established pooling methods.¹

1 Introduction

Dowker homology (DH; Dowker, 1952; Chowdhury and Mémoli, 2018) is a tool from topological data analysis, an emerging data analysis framework drawing on ideas from topology, metric geometry, and related areas of mathematics. Among other things, DH may be used to capture the relative position of two point clouds living in a common space that is endowed with a distance function (typically $\mathbb{R}^n$ with Euclidean or cosine distance).

In this paper, we study a novel application of DH to NLP by applying it to Siamese sentence transformer models (Reimers and Gurevych, 2019). Specifically, we ask the following.

- (1) Can DH distinguish similar and dissimilar sentence pairs?

- (2) Can DH be used to extract sentence similarity from token embeddings?

On the positive side, we find that DH can elucidate sentence similarity data and models, both quantitatively and visually. Findings regarding the second question are mixed: we define single-number summaries extracted from DH that capture ground truth sentence similarity reasonably well, but not more so than established methods.

2 Background on Dowker Homology

This section provides a high-level review of DH; for details, we refer the reader to Appendix A.1. We limit ourselves to reviewing zero-dimensional DH, as that is the version of DH used in our experiments.

We consider a pair of point clouds X_1 and X_2 living in a common space that is endowed with a distance function d . We say that a point in X_1 or X_2 is r -close to another if the distance between the two points is at most r (as measured by d). Given a scale $r \geq 0$, one constructs a graph whose vertices are those points in X_1 that have a neighbor in X_2 that is r -close, and where two points in X_1 are connected by an edge whenever there exists a point in X_2 that is r -close to *both* of them. In particular, the graph is constructed on elements of X_1 only, while X_2 merely dictates the connectivity of the graph. To obtain DH of X_1 relative to X_2 , one initially sets the scale to $r = 0$, and gradually increases it to theoretical infinity. Along the way, one keeps track of the number of connected components constituting the aforementioned graph: any such component “is born” at a certain scale, and “dies” at some later scale as it is absorbed by—that is, connected via an edge to—an existing component. This yields a sequence $(b_1, d_1), \dots, (b_k, d_k)$, where each pair contains the scales at which the birth and death, respectively, of a component happen. This sequence is recorded in a *persistence*

¹All code to run experiments is available at <https://github.com/m-a-huber/dowker-for-sent-sim> and archived at <https://doi.org/10.5281/zenodo.21810253>.

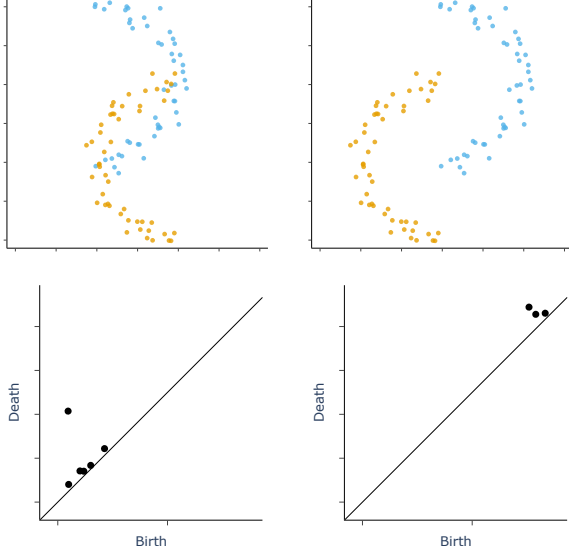


Figure 1: Two pairs of point clouds of varying degree of spatial separation (top) and their corresponding Dowker persistence diagrams (bottom). The more the point clouds are separated, the larger the birth values in the persistence diagrams.

diagram, which is obtained by placing a point at coordinates (b_i, d_i) in $\mathbb{R}^2$ for all $i = 1, \dots, k$.² A key property of DH is that it is unaffected when swapping the roles of X_1 and X_2 in the above process, making the resulting persistence diagram an “undirected” quantity (Dowker, 1952; Chowdhury and Mémoli, 2018).

If the point clouds X_1 and X_2 are spatially well separated, vertices and edges appear only at large scales. If X_1 and X_2 nearly coincide, those appear at small scales already; indeed, if $X_1 = X_2$, all birth values equal zero. The spatial proximity (resp. separation) of X_1 and X_2 is thus reflected in the points of the Dowker persistence diagram having small (resp. large) birth values; see Figure 1.

3 Dowker Homology for Similarity

To apply DH to sentence similarity, we treat the token embeddings stemming from a pair of sentences as a pair of point clouds in the latent space of a transformer model, and compute the corresponding Dowker persistence diagram. If the two sentences are similar (resp. dissimilar), we expect the corresponding point clouds to be colocalized (resp. separated) and, as discussed in Section 2, we

²A persistence diagram is usually endowed with the diagonal line where birth values equal death values. The vertical distance of the i -th point to the diagonal equals $d_i - b_i$, and can thus be thought of as the “lifetime” of the corresponding connected component, $i = 1, \dots, k$.

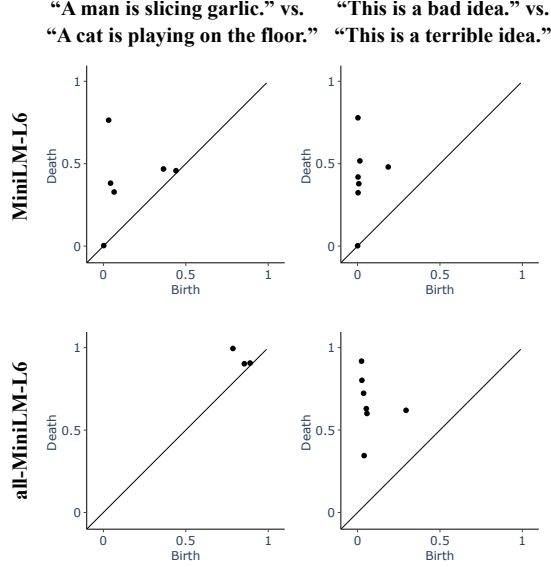


Figure 2: Dowker persistence diagrams capturing sentence similarity of two randomly selected sentence pairs from the STS-B dataset with low (left column) and high (right column) similarity, computed from embeddings produced using MiniLM-L6 (top row) and all-MiniLM-L6 (bottom row). Higher sentence similarity is reflected in smaller birth values in the diagrams, and the fact that MiniLM-L6 is not fine-tuned for sentence similarity is reflected in the top two diagrams being more similar to each other than the bottom two diagrams. Moreover, fine-tuning MiniLM-L6 for sentence similarity substantially alters the persistence diagram corresponding to the dissimilar sentence pair, while leaving the other diagram largely unaffected.

expect the resulting Dowker persistence diagram to have predominantly small (resp. large) birth values.

We point out that our method is similar in spirit to those used in Barannikov et al. (2022) and Michel et al. (2017). Indeed, both methods also draw on ideas from topological data analysis to assess the (dis-)similarity of two point clouds. The former method, however, requires the two point clouds to be of equal sizes, while the latter takes into account only the topological features of the two point clouds individually and does not capture, for instance, their spatial separation.

Example. Figure 2 illustrates that our expectation holds for two sentence pairs randomly selected from the STS Benchmark (STS-B; Cer et al., 2017) when using the fine-tuned all-MiniLM-L6 model, but not when using the non-fine-tuned MiniLM-L6 model. Fine-tuning a transformer model for sentence similarity seems to strongly affect how dissimilar sentences are represented (as seen through the lens of DH), but less so for similar sentences,

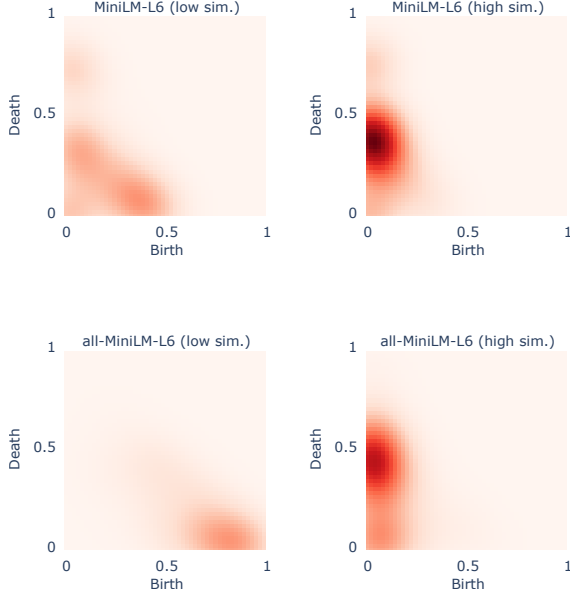


Figure 3: Mean persistence images capturing sentence similarity of all sentence pairs from the STS-B dataset with low (left column) and high (right column) similarity, computed from embeddings produced using MiniLM-L6 (top row) and all-MiniLM-L6 (bottom row). These persistence images can be thought of as “heat maps” that capture the distribution of points across multiple persistence diagrams (see Appendix A.2 for details). The images indicate that fine-tuning MiniLM-L6 for sentence similarity leads to larger birth values in DH of dissimilar sentence pairs, thus generalizing the findings from Figure 2 across the entire STS-B dataset. Here, the sentence pairs with low (resp. high) similarity are those whose ground truth similarity in STS-B differs by at most 0.1 from the minimum (resp. maximum) possible similarity value. For the analogous plots for the other models used, see Appendix A.5.1.

where the persistence diagrams stemming from the non-fine-tuned and the fine-tuned model are basically indistinguishable. Indeed, the finding of Figure 2 generalizes across all similar and dissimilar pairs from the STS-B dataset, as illustrated in Figure 3.

Experiment outline. Motivated by this example, we conduct the following two experiments for each layer of every model from a diverse set of transformer models fine-tuned for sentence similarity as well as their underlying, non-fine-tuned base models (see Table A.1 in Appendix A.3 for a list of models used). We use sentence pairs from the STS-B dataset, which contains sentence pairs annotated by human raters with semantic similarity scores ranging from 0 to 5.

1. Layer regression experiment: We investi-

gate to what extent standard token-level measures as well as Dowker persistence diagrams contain information on sentence similarity. We do so by fitting a linear regression model that predicts ground truth sentence similarities from these measures.

2. Layer correlation experiment: In order to circumvent the need for a regression trained on Dowker persistence diagrams, and to provide measures of sentence similarity that are more readily applicable than the raw persistence diagrams, we define single-number summaries extracted from DH. These summaries are defined so that we expect them to capture sentence similarity directly, and we assess how well they correlate with ground truth sentence similarities.

We now describe the sentence similarity methods used in our experiments.

3.1 Baseline Similarity Methods

We use BOS-, EOS-, max- and mean-pooling as single-vector sentence representations. The similarity of a sentence pair is then computed as the cosine similarity between the pooled representations of the sentences.

3.2 Dowker Similarity

We define four versions of *Dowker similarity* (DS), one for each of four different kinds of aggregation used along the way. To compute DS for a sentence pair (s_1, s_2) , we consider the token embeddings of s_1 and s_2 as point clouds X_1 and X_2 , respectively, and compute DH of X_1 relative to X_2 , as explained in Section 2. We extract single-number summaries from the resulting persistence diagram by computing the minimum, maximum and average of all birth values of points in the diagram, as well as the weighted average birth value, where birth values are weighted by the lifetime of the corresponding point. We regard each of these as a notion of distance between X_1 and X_2 , since larger birth values indicate a larger degree of spatial separation of X_1 and X_2 (see Section 2), and we denote these distances by $d_{\text{Dow}}^{\min}$, $d_{\text{Dow}}^{\max}$, $d_{\text{Dow}}^{\text{mean}}$ and $d_{\text{Dow}}^{\text{wght}}$, respectively. Finally, we define the DSs by setting $s_{\text{Dow}}^{\bullet}(s_1, s_2) = e^{-d_{\text{Dow}}^{\bullet}(X_1, X_2)}$ for $\bullet \in \{\min, \max, \text{mean}, \text{wght}\}$. All versions of DS thus attain values between 0 and 1. Moreover, if $s_1 = s_2$, then $X_1 = X_2$, and all birth values in DH equal zero, so that all four versions of Dow-

ker distance vanish, and, correspondingly, all four versions of DS yield a similarity value of 1 for identical sentences.

4 Experiments and Results

We now provide detailed descriptions of the two experiments conducted, and then briefly discuss some limitations and design choices.

4.1 Layer Regression Experiment

Setup. We sample 512 and 256 sentence pairs from the train and test splits, respectively, of STS-B. As baseline features, we use the vectors $(u_1, u_2, |u_1 - u_2|)$, where u_1 and u_2 are the pooled representations of a sentence pair (s_1, s_2) , obtained via the four pooling methods described in Section 3.1 (this is motivated by Reimers and Gurevych (2019, Section 6); here, $|\cdot|$ denotes the entry-wise absolute value). For DH, we compute the persistence diagram of each sentence pair from the corresponding token embeddings (as described in Section 3.2) using cosine distance, and vectorize it into a persistence image of resolution 25×25 , resulting in a vector of length 625 (see Appendix A.2 for details on persistence images). We then train a ridge regression on these features and evaluate it using Spearman’s rank correlation between the predicted and the ground-truth similarities; the choice of ridge regression is motivated by the fact that persistence images produce high-dimensional, sparse vectors with correlated features, making ordinary least squares estimates potentially unstable (see Appendix A.4 for selection of hyperparameters). This procedure is run for each hidden layer of each model.

Results. We find that across all models (fine-tuned or not), the regression trained on the Dowker persistence diagrams never outperforms the one trained on the best performing baseline similarity methods. The regression trained on the Dowker persistence diagrams always wins at least third place (as measured by Spearman correlation averaged across all layers of a model), consistently beating those trained on BOS- and EOS-pooling, but coming short of outperforming those trained on max- and mean-pooling. Interestingly, max-pooling yields the best features for 9/10 models despite the fact that the sentence transformers used were fine-tuned using mean-pooling. Finally, we observe that in the fine-tuned models—in contrast to the non-fine tuned ones—all similarity methods

tend to yield higher Spearman correlations on later layers, which is somewhat intuitive as the final layer is where fine-tuning is directly applied. As an example, Figure 4 shows the Spearman correlations of all methods at each layer of MiniLM-L6 (not fine-tuned) and all-MiniLM-L6 (fine-tuned); the former is one of the two cases in which the features derived from DH rank second, outperforming those derived from mean-pooling. Appendix A.5.2 shows analogous plots for the other model pairs.

4.2 Layer Correlation Experiment

Setup. We sample 256 sentence pairs from the test split of STS-B, and compute similarities for each hidden layer of each model, using the four baseline similarity methods and the four versions of DS from Sections 3.1 and 3.2, respectively. For each method, model and layer, we evaluate the performance via Spearman correlation between the predicted and the ground-truth similarities.

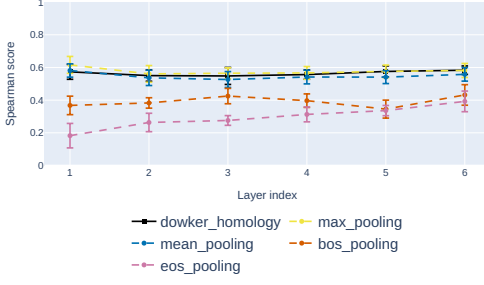
Results. As before, we assess the goodness of the baseline methods and the four versions of DS by the Spearman correlations averaged across all layers of a model. We find that (at least one version of) DS comes in third behind the max- and mean-pooling methods for 5/10 models, and wins first and second place for 2/10 models each. Again, we find that max-pooling outperforms mean-pooling in terms of average Spearman correlation. Mean-pooling, however, unsurprisingly yields the highest maximal Spearman correlation for the sentence transformer models, and this is always the case at the final model layer, sometimes after a sheer increase of performance in the last few layers. Finally, we find that $s_{\text{Dow}}^{\text{mean}}$ outperforms the other versions of DS for 9/10 models. As an example, Figure 5 shows the Spearman correlations of all methods at each layer of MiniLM-L6 and all-MiniLM-L6; the former is one of two cases where $s_{\text{Dow}}^{\text{mean}}$ outperforms all other methods (the other such instance being MiniLM-L12). We refer to Appendix A.5.3 for plots of the remaining results.

4.3 Experiment Limitations

Dataset and Sample Sizes. In both experiments, we confined ourselves to working with just the STS-B dataset and the relatively modest respective sample sizes due to computational constraints.

Computation of DH Feature Vectors. While DH may be computed using any metric, we restricted ourselves to using cosine distance in our

Layer-wise ridge regression for MiniLM-L6:
(Spearman mean & SD across 8 repetitions)



Layer-wise ridge regression for all-MiniLM-L6:
(Spearman mean & SD across 8 repetitions)

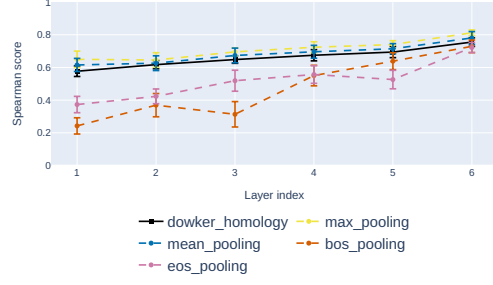
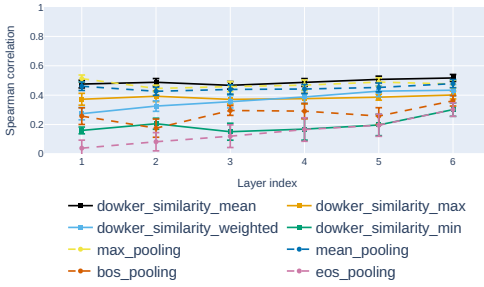


Figure 4: Two results of the layer regression experiment. Bars indicate standard deviation over eight runs.

Layer correlation for MiniLM-L6:
(Spearman mean & SD across 8 repetitions)



Layer correlation for all-MiniLM-L6:
(Spearman mean & SD across 8 repetitions)

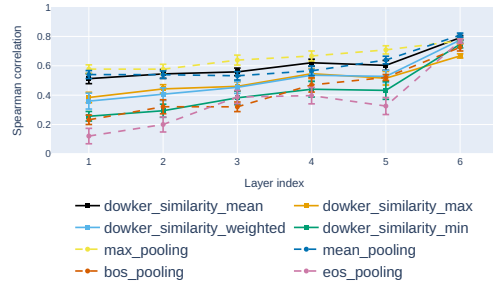


Figure 5: Two results of the layer correlation experiment. Versions of Dowker similarity are indicated by the suffixes.

experiments. We experimented using Euclidean distance, but cosine distance consistently yielded better results. For the layer regression experiment, the resolution of 25×25 used for vectorization of persistence diagrams was chosen as an empirical compromise between preserving performance and limiting the dimensionality of the resulting feature vectors. Finally, we restricted ourselves to working with zero-dimensional DH because we suspected higher-dimensional DH to capture information that goes beyond spatial separation of point clouds. Indeed, preliminary experiments showed that the point clouds consisting of token embeddings rarely exhibited any higher-dimensional DH at all, and including higher-dimensional DH as features in the layer regression experiment had virtually no effect on the results.

Evaluation metric. In both experiments, we confined ourselves to evaluating the agreement of DSs with ground truth similarities via Spearman correlation since DSs are obtained by applying the monotonic transformation $d \mapsto e^{-d}$ to the uncalibrated distance measures $d_{\text{Dow}}^{\min}$, $d_{\text{Dow}}^{\max}$, $d_{\text{Dow}}^{\text{mean}}$ and

$d_{\text{Dow}}^{\text{wght}}$. We hence do not expect any linear relationship between DSs and ground truth similarities as captured, for instance, by Pearson correlation.

5 Discussion

Our experiments show that Dowker persistence diagrams reflect sentence similarity (cf. Question (1)) and, moreover, that they can provide visual insights on NLP models, such as the effect of fine-tuning on the representation of similar and dissimilar sentence pairs (cf. the example in Section 3). Moreover, while all versions of DS capture sentence similarity to some extent, the version that averages birth values is by far the best-performing one, as witnessed by the fact that it correlates with ground truth similarities essentially as well as a trained linear regression. Even this version of DS, however, does not consistently outperform established methods, answering Question (2) with a “yes, but”. In summary, our findings provide insights on the usefulness and interpretability of DH, and justify further research on connections between topology and sentence similarity, such as exploration of further single-number summaries extracted from DH.

References

- Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. 2017. [Persistence images: A stable vector representation of persistent homology](#). *Journal of Machine Learning Research*, 18(8):1–35.
- Serguei Barannikov, Ilya Trofimov, Nikita Balabin, and Evgeny Burnaev. 2022. [Representation topology divergence: A method for comparing neural network representations](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1607–1626. PMLR.
- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. [SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.
- Samir Chowdhury and Facundo Mémoli. 2018. [A functorial Dowker theorem and persistent homology of asymmetric networks](#). *J. Appl. Comput. Topol.*, 2(1-2):115–175.
- Pawel Dlotko. 2026. [Persistence representations](#). In *GUDHI User and Reference Manual*, 3.12.0 edition. GUDHI Editorial Board.
- C. H. Dowker. 1952. [Homology groups of relations](#). *Ann. of Math. (2)*, 56:84–95.
- Marius Huber, David Robert Reich, and Lena Ann Jäger. 2026. [Fixation sequences as time series: A topological approach to dyslexia detection](#). In *Proceedings of the 2026 Symposium on Eye Tracking Research and Applications, ETRA ’26*, New York, NY, USA. Association for Computing Machinery.
- Marius Huber and Patrick Schnider. 2025. [Flagifying the dowker complex](#). *Preprint*, arXiv:2508.08025. ArXiv:2508.08025 [math.AT], <https://arxiv.org/abs/2508.08025>.
- Paul Michel, Abhilasha Ravichander, and Shruti Rijhwani. 2017. [Does the geometry of word embeddings help document classification? A case study on persistent homology-based representations](#). In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 235–240, Vancouver, Canada. Association for Computational Linguistics.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 Dowker Homology

DH, introduced by Dowker (1952) and promoted to the setting of persistent homology by Chowdhury and Mémoli (2018), is a tool from topological data analysis that captures the topology of two sets X and Y and a relation $R \subseteq X \times Y$ between them. Here, we review DH as used in the main text of the article. We point out that the implementation of DH used in our experiments is that of Dowker–Rips homology provided by Huber and Schnider (2025). While Dowker–Rips homology is, in general, merely an approximate, computationally cheaper variant of DH, we use that implementation since all our experiments rely only on zero-dimensional DH, in which case Dowker and Dowker–Rips homology coincide. Correspondingly, the following exposition is restricted to zero-dimensional DH.

In the case where X and Y are subsets of some metric space (Z, d) , and given some scale $r \geq 0$, one may define a relation R_r by declaring that $(x, y) \in R_r$ iff $d(x, y) \leq r$, for $x \in X$ and $y \in Y$. The relation R_r may be translated into the graph $\Gamma_r = (V_r, E_r)$, defined as having vertex set

$$V_r = \{x \in X \mid \exists y \in Y: d(x, y) \leq r\} \subseteq X,$$

and where two vertices $x_1, x_2 \in X$ are connected via an edge whenever there exists $y \in Y$ such that $d(x_1, y) \leq r$ and $d(x_2, y) \leq r$.

As is the case for any graph, the graph Γ_r consists of some—possibly zero—connected components. Indeed, if $r = 0$, $\Gamma_r = \emptyset$ (provided that X and Y have no common elements). As one increases the scale r , vertices and edges appear in Γ_r . Correspondingly, the number of connected components of Γ_r can increase or decrease as r increases: a new vertex may appear when r becomes large enough for that vertex to “see” an element of Y , or two existing components of Γ_r may merge into one when r becomes large enough for two vertices initially belonging to different connected components to be connected via an edge. In other

words, increasing the scale r from zero to (theoretical) infinity yields a sequence “births” and “deaths” of connected components. DH is the information contained in this process, and may be recorded by a sequence $(b_1, d_1), \dots, (b_k, d_k)$ of pairs of non-negative real numbers, where the i -th pair contains the value of the scale r at which the “birth” (b_i) and “death” (d_i), respectively, of the i -th connected component occur.

This sequence is usually recorded in a *persistence diagram*, which, by definition, is the subset of $\mathbb{R}^2$ consisting of the pairs $(b_1, d_1), \dots, (b_k, d_k)$ (see Figure 1 in the main text for examples of persistence diagrams). Persistence diagrams are usually endowed with the diagonal line characterized by those points where birth values equal death values; the vertical distance of the point (b_i, d_i) in the persistence diagram indicates the lifetime (that is, $d_i - b_i$) of the i -th connected component. As such, a persistence diagram provides an at-a-glance summary of the evolution of the graph Γ_r as r increases from zero to infinity, and—as explained in Section 2—captures the spatial separation of X and Y as measured by the metric d defined on the metric space (Z, d) . We close this section by remarking that any Dowker persistence diagram contains precisely one point whose lifetime is infinite, that is, which dies at only scale $r = \infty$. Since this infinite value typically breaks many formulas and pipelines, one usually either discards this point from the diagram, or replaces the scale at which it dies by some finite number that is chosen according to some heuristic. In our experiments, we experimented with both options, and found that simply discarding that point yielded better and more stable results.

A.2 Vectorizing Persistence Diagrams

To make persistence diagrams usable in machine learning pipelines, one must transform them into vectors of a fixed length. In our experiments, we use a method known as *persistence image* (Adams et al., 2017). A persistence image is obtained from a persistence diagram by first applying a shearing that maps the diagonal of the persistence diagram to a horizontal line. The resulting diagram is then converted into a “heat map” by placing a bivariate Gaussian distribution with a given standard deviation (also referred to as *bandwidth* in the present context) at each point of the diagram. This yields a single-channel image commonly referred to as persistence surface; see Figure A.1 for an illus-

tration of the process of obtaining a persistence surface from a persistence diagram. Finally, the persistence image is obtained from the persistence surface by discretizing it into a pixel grid of a chosen *resolution*. The bandwidth and resolution used in the process are treated as hyperparameters in applications.

A.3 Models used

See Table A.1 for a list of models used in our experiments.

A.4 Hyperparameter Selection

To train the ridge regressions used in the layer regression experiment described in Section 4.1, we used the implementation provided by scikit-learn (Pedregosa et al., 2011). In doing so, we used the SVD-solver, tuned the regularization parameter α via grid search over the values $10^{-3}, 10^{-2}, \dots, 10^2$, and left other parameters at their default values. Persistence diagrams in these experiments were transformed into persistence images using the GUDHI library (Dlotko, 2026). We used default values for persistence image creation, apart from the bandwidth, which we tuned via grid search over the values $10^{-3}, 10^{-2}, \dots, 10^2$.

A.5 Results

Here we provide plots containing the results from the two experiments conducted, analogous to the plots in the main text.

A.5.1 Mean Persistence Images for Dissimilar and Similar Sentence Pairs

See Figure 3 in the main text and Figures A.2a to A.2d.

A.5.2 Layer Regression Experiment

See Figure 4 in the main text and Figures A.3 to A.6.

A.5.3 Layer Correlation Experiment

See Figure 5 in the main text and Figures A.7 to A.10.

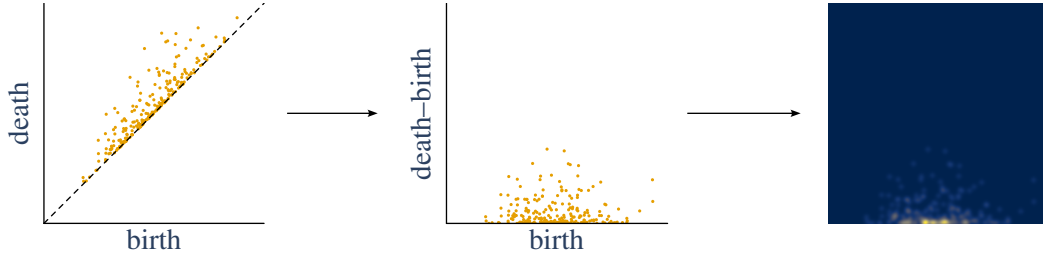
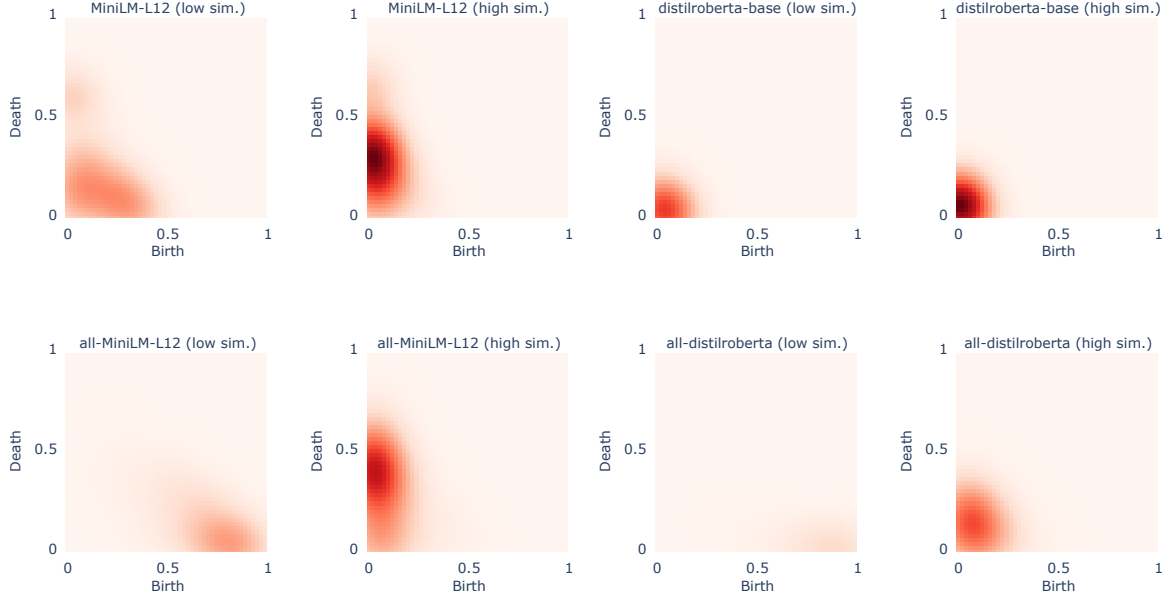


Figure A.1: Schematic showing the transformation of a persistence diagram in (birth, death)-coordinates (left) to (birth, death-birth)-coordinates (middle) by means of a shearing map, and the resulting persistence surface (right). Figure adapted from [Huber et al. \(2026\)](#).

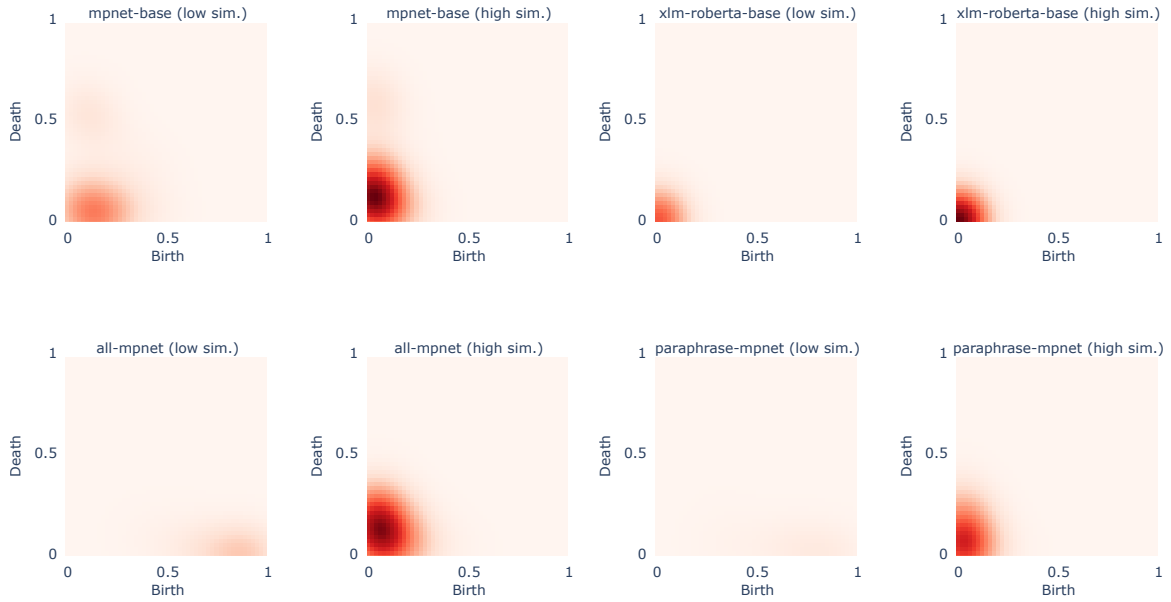
Base model	Sentence transformer
MiniLM-L6-H384-uncased	all-MiniLM-L6-v2
MiniLM-L12-H384-uncased	all-MiniLM-L12-v2
distilroberta-base	all-distilroberta-v1
mpnet-base	all-mpnet-base-v2
xlm-roberta-base	paraphrase-multilingual-mpnet-base-v2

Table A.1: Sentence transformers and respective base models used in our experiments.



(a) Mean persistence images for the models MiniLM-L12-H384-uncased and all-MiniLM-L12-v2.

(b) Mean persistence images for the models distilroberta-base and all-distilroberta-v1.

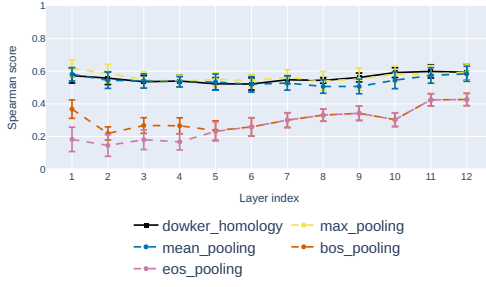


(c) Mean persistence images for the models mpnet-base and all-mpnet-base-v2.

(d) Mean persistence images for the models xlm-roberta-base and paraphrase-multilingual-mpnet-base-v2.

Figure A.2: Mean persistence images for the model pairs from Table A.1. See Figure 3 in the main text for a description of what these images capture and how they are obtained.

Layer-wise ridge regression for MiniLM-L12:
(Spearman mean & SD across 8 repetitions)



Layer-wise ridge regression for all-MiniLM-L12:
(Spearman mean & SD across 8 repetitions)

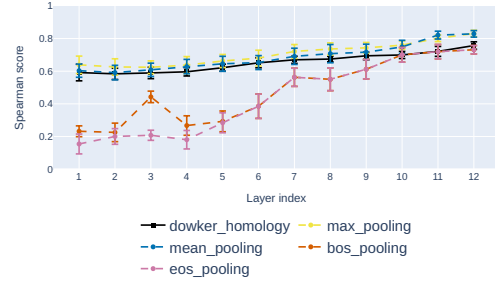
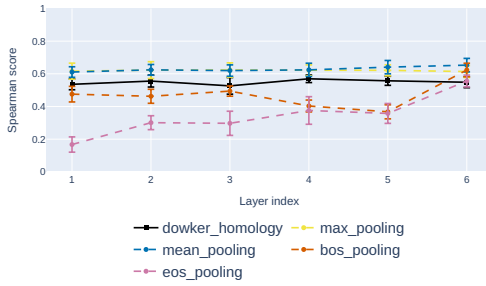


Figure A.3: Results of the layer regression experiment for MiniLM-L12-H384-uncased (left) and all-MiniLM-L12-v2 (right).

Layer-wise ridge regression for distilroberta-base:
(Spearman mean & SD across 8 repetitions)



Layer-wise ridge regression for all-distilroberta:
(Spearman mean & SD across 8 repetitions)

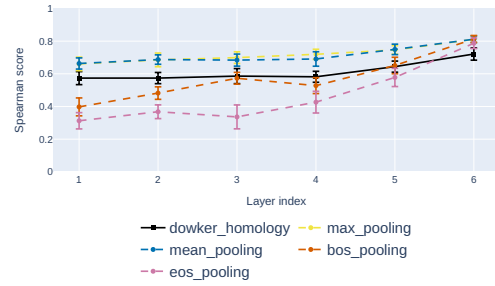
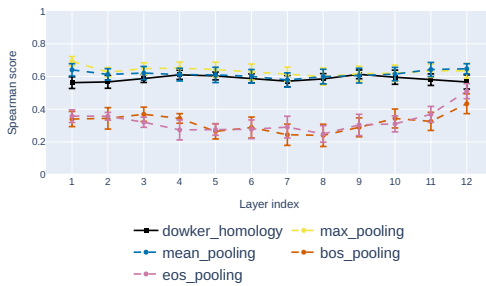


Figure A.4: Results of the layer regression experiment for distilroberta-base (left) and all-distilroberta-v1 (right).

Layer-wise ridge regression for mpnet-base:
(Spearman mean & SD across 8 repetitions)



Layer-wise ridge regression for all-mpnet:
(Spearman mean & SD across 8 repetitions)

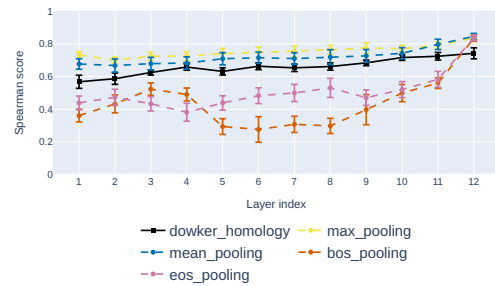
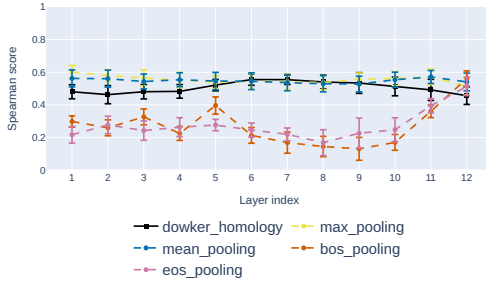


Figure A.5: Results of the layer regression experiment for mpnet-base (left) and all-mpnet-base-v2 (right).

Layer-wise ridge regression for xlm-roberta-base:
(Spearman mean & SD across 8 repetitions)



Layer-wise ridge regression for paraphrase-mpnet:
(Spearman mean & SD across 8 repetitions)

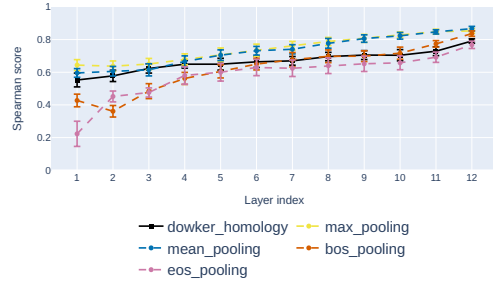
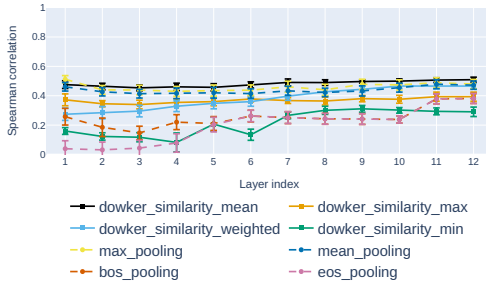


Figure A.6: Results of the layer regression experiment for xlm-roberta-base (left) and paraphrase-multilingual-mpnet-base-v2 (right).

Layer correlation for MiniLM-L12:
(Spearman mean & SD across 8 repetitions)



Layer correlation for all-MiniLM-L12:
(Spearman mean & SD across 8 repetitions)

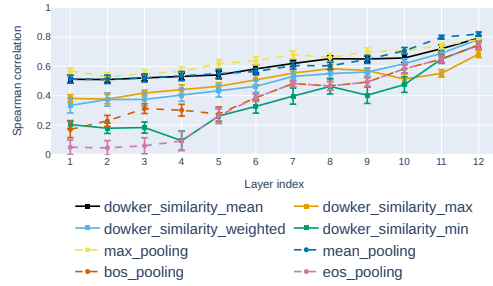
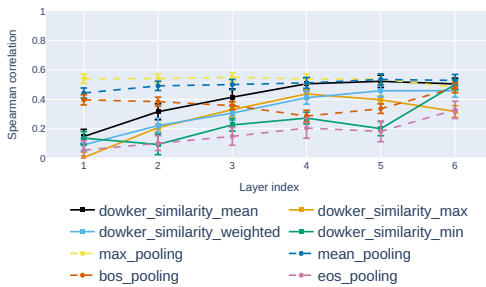


Figure A.7: Results of the layer correlation experiment for MiniLM-L12-H384-uncased (left) and all-MiniLM-L12-v2 (right). Method names in the legend are sorted according to decreasing Spearman correlation averaged across model layers.

Layer correlation for distilroberta-base:
(Spearman mean & SD across 8 repetitions)



Layer correlation for all-distilroberta:
(Spearman mean & SD across 8 repetitions)

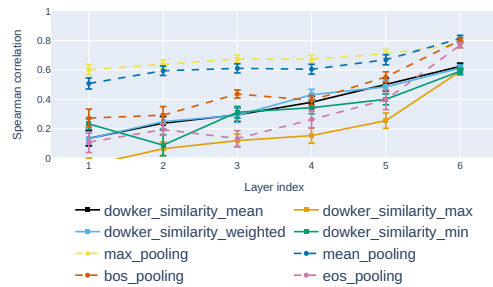
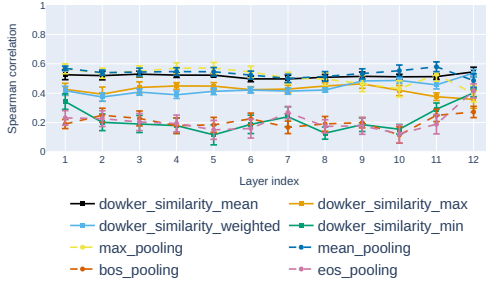


Figure A.8: Results of the layer correlation experiment for distilroberta-base (left) and all-distilroberta-v1 (right). Method names in the legend are sorted according to decreasing Spearman correlation averaged across model layers.

Layer correlation for mpnet-base:
(Spearman mean & SD across 8 repetitions)



Layer correlation for all-mpnet:
(Spearman mean & SD across 8 repetitions)

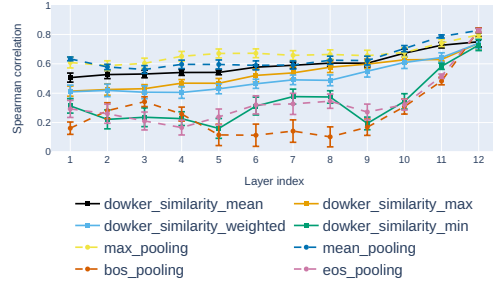
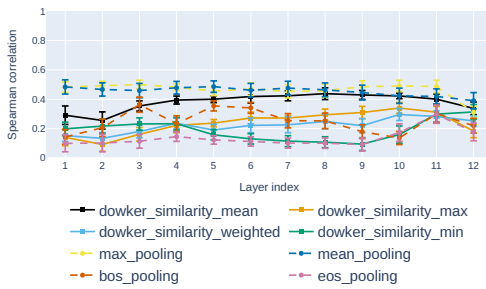


Figure A.9: Results of the layer correlation experiment for mpnet-base (left) and all-mpnet-base-v2 (right). Method names in the legend are sorted according to decreasing Spearman correlation averaged across model layers.

Layer correlation for xlm-roberta-base:
(Spearman mean & SD across 8 repetitions)



Layer correlation for paraphrase-mpnet:
(Spearman mean & SD across 8 repetitions)

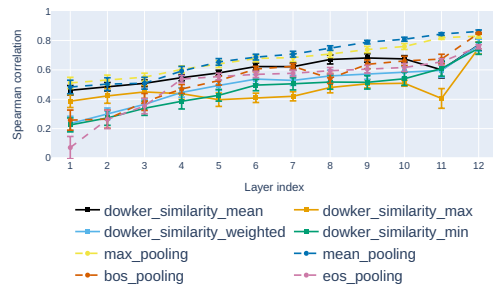


Figure A.10: Results of the layer correlation experiment for xlm-roberta-base (left) and paraphrase-multilingual-mpnet-base-v2 (right). Method names in the legend are sorted according to decreasing Spearman correlation averaged across model layers.

Does Pragmatic Transfer Help Euphemism Detection? Evidence from Target-Span Marking

Whitney Poh and Noriko Takahashi and Patrick Lee and Julia Sammartino
and Libby Barak and Jing Peng and Anna Feldman

Montclair State University

New Jersey, USA

{pohw1,takahashin1,leep,sammartinoj,barakl,pengj,feldmana}@montclair.edu

Abstract

Euphemism detection is often motivated as a pragmatic task: deciding whether a potentially euphemistic term (PET) is used euphemistically depends on context, communicative goals, politeness, topic sensitivity, and sentiment. A natural prediction is that auxiliary supervision from pragmatically related tasks should improve euphemism detection when target supervision is limited. We test this prediction for PET-style classification using XLM-R and mBERT in a feature-based transfer setting: auxiliary-tuned encoders are frozen and probed for euphemism detection. We compare sentence-level pragmatic proxy tasks with expression-level nonliteral tasks and with target inputs that either mark or omit the PET span. The results do not support a simple pragmatic-transfer account. Auxiliary transfer is unreliable, often reducing XLM-R performance and helping mBERT only inconsistently. The largest observed gains come from direct euphemism fine-tuning when the target PET is explicitly marked. The practical implication is to preserve target-span information and report source–target compatibility in transfer experiments.

1 Introduction

Euphemisms replace direct or potentially offensive, negative or socially taboo expressions with more socially acceptable alternatives (Lee et al., 2024; Poh et al., 2026; Firsich and Rios, 2024, etc.). Because euphemistic meaning depends on context and communicative goals, euphemisms are commonly linked to pragmatic dimensions such as politeness (Gavidia et al., 2022; Lee et al., 2024), topic sensitivity (Gavidia et al., 2022; Lee et al., 2023), and sentiment (Felt and Riloff, 2020). We ask whether encoders exposed to these pragmatic signals through auxiliary tasks become better eu-

phemism detectors.¹

PET-style disambiguation requires interpreting a specific expression in context. For example, *let go* is euphemistic in “the company let most of its employees go”, referring to being fired, but literal in “let go of my hand”. Likewise, *reorganizing* may refer to layoffs or tidying one’s room, and *slept with* may imply a sexual relationship or literal co-sleeping. Several examples are multiword expressions (MWE), but MWE status is not the hypothesis being tested. An example of a non-MWE euphemism found in the dataset by Lee et al. (2024) is “late”, meaning dead. The relevant property is that the label concerns a particular expression in context.

The pragmatic-transfer prediction is not supported in our experiments, which utilize XLM-R (Conneau et al., 2020) and mBERT (Devlin et al., 2019). Transfer from politeness, sentiment, topic sensitivity, and related proxy tasks is unreliable: it is usually negative for XLM-R and inconsistent for mBERT. This result is informative because these auxiliary tasks are not arbitrary baselines; they are plausible choices from a linguistic perspective. Euphemisms are pragmatic, but sentence-level pragmatic supervision does not necessarily teach a model how to decide whether a specific expression is euphemistic in a given context.

Rather than viewing the transfer result solely as evidence about the linguistic properties of euphemisms, we treat it as an opportunity to examine the relationship between task structure and transfer learning. Expression-level classification tasks, including many multiword-expression and figurative-language tasks, often use boundary marking to make a decision about a specific target span embedded within a larger context (Gavidia et al., 2022; Lee et al., 2024). In such settings, per-

¹Code and other material available at <https://github.com/NLP1abMSU/Cross-Task>

formance may depend not only on the linguistic phenomenon being modeled, but also on whether the task explicitly identifies the expression under consideration. We therefore compare inputs with and without boundary markers around the target expression and examine whether auxiliary tasks that require span-conditioned decisions transfer differently from tasks based on sentence-level or discourse-level properties.

Our contributions are threefold. First, we report a negative result: pragmatically related auxiliary tasks do not reliably transfer to PET-style euphemism detection. Second, we show that explicit target-span marking creates larger and more stable gains compared to unmarked versions of the same auxiliary task, and that factors such as target localization, label semantics, and input format may impact results significantly, shaping what information models learn and whether auxiliary supervision becomes useful. Direct euphemism fine-tuning in which both the training and testing set contains euphemism boundaries produced the highest and most consistent gains. Third, we derive a resource-design recommendation: when labels concern a specific expression, datasets should preserve target-span information because removing it changes both task difficulty and the interpretation of transfer results.

2 Background

Recent work has expanded euphemism detection across languages and domains using PET-style datasets (Gavidia et al., 2022; Lee et al., 2023, 2024; Biyik et al., 2024; Lee and Feldman, 2024). Methods include lexicon-based approaches (Felt and Riloff, 2020; Lee et al., 2022), transformer-based classifiers (Zhu et al., 2021; Wang et al., 2022), and prompting-based evaluations (Firsich and Rios, 2024). PET-style datasets emphasize contextual disambiguation: the same expression may be euphemistic in one context and literal in another.

Euphemisms also serve different communicative functions, including face-saving, softening and strategic obfuscation (Lee et al., 2024), such as in political discussions about climate change (Grolleau et al., 2022). Euphemistic and literal expressions can be semantically close while differing in pragmatic impact and social consequences (Grolleau et al., 2022). This motivates pragmatic auxiliary tasks, but it also raises a methodological question: does broad pragmatic relatedness transfer, or

does the auxiliary task need to match the target task’s decision unit?

Transfer learning can help low-resource target tasks, but it can also hurt through negative transfer (Wang et al., 2019; Chen et al., 2024). Prior work shows that transferability depends on task compatibility, supervision format, and training order (Vu et al., 2020). Our experiments test a concrete negative-transfer hypothesis: auxiliary tasks that appear linguistically related to euphemism detection may still be unhelpful when their input structure and decision unit differ from the PET-style target task.

3 Data

Target task. We use the English subset of the PET dataset released by Lee et al. (2024); Lee and Feldman (2024), containing 3,098 instances: 1,841 euphemistic and 1,257 non-euphemistic. Each instance contains a PET whose usage must be classified. We create two target variants: EUPH-B, where boundary tokens surround the PET, and EUPH-NOB, where the same texts are used without boundary markers. The marker appears in all positive and negative examples, so it identifies the target span but does not encode the class label. For euphemisms, sampling is restricted to `sometimes_euph` expressions, i.e., expressions appearing with both euphemistic and non-euphemistic labels, to reduce PET-identity memorization.

Auxiliary tasks. We evaluate eleven auxiliary input conditions drawn from eight auxiliary families (Table 1). The auxiliary datasets are drawn from established resources: Stanford Politeness (Danescu-Niculescu-Mizil et al., 2013), Sensitivity following Lee et al. (2023) and Mejova et al. (2014) using the GloWbe corpus (Davies and Fuchs, 2015), RT Movie Sentiment (Pang and Lee, 2005), Books vs. Electronics reviews (Blitzer et al., 2007), Magpie (idioms) (Haagsma et al., 2020; Dankers et al., 2022), TroFi (metaphors) (Birke and Sarkar, 2007), and IDEM (Idiom Sentiment) (Prochnow et al., 2024)². The pragmatic proxies are politeness, sensitivity, and sentiment. Books vs. Electronics is a domain-control task. Magpie and TroFi are expression-level nonliteral-language tasks included to test decision-unit and span-format alignment. We create boundary and

²See Appendix A for more details on datasets and source URLs

Task	Unit	Role
PET	target expr.	target
Politeness	utterance	prag. proxy
Sensitivity	topic	topic proxy
Sentiment	review	sent. proxy
Books/Elect.	review	control
Magpie	target expr.	expr.-level
TroFi	target word	expr.-level
IDEM	idiom sent.	affective
Multi	mixed	broad exposure

Table 1: Auxiliary task families and their primary decision units.

no-boundary variants for Magpie and TroFi by inserting or removing the same marker used in the PET data. IDEM tests whether affective supervision over idiom-containing sentences helps without requiring literal/nonliteral disambiguation. Mixed-task samples (Multi = Movie Sentiment (Pang and Lee, 2005), Sensitivity (Lee et al., 2023), Magpie (boundaries) (Haagsma et al., 2020; Dankers et al., 2022) and Politeness (Danescu-Niculescu-Mizil et al., 2013)) test whether broader exposure improves transfer beyond the best single auxiliary source; because their labels retain source-specific meanings, they are treated as broad-exposure controls rather than semantically unified tasks.

Examples of utterances from auxiliary sets can be found in Table 12, examples of euphemistic statements with certain pragmatic traits can be found in Table 13.

To reduce trivial effects of dataset size, and to standardize class balance for each setup, we sample 950 instances per class for each task, yielding 1,900 training instances per condition. Instances longer than the maximum length are removed during tokenization. For Sensitivity, which is drawn from GloWbE (Davies and Fuchs, 2015), which the Euphemisms dataset also draws from, we remove overlapping instances to reduce cross-task contamination.

4 Experimental Setup

4.1 XLM-R/mBERT

Our transfer setting is feature-based. We first fine-tune an encoder on an auxiliary task. We then freeze the encoder, extract final-layer classification-token embeddings for the euphemism data, and train a logistic regression probe on the euphemism training split. This isolates representational compatibility rather than full end-to-end adaptation. Fine-tuning and inference were done using HuggingFace Transformers (Wolf et al., 2020).

We use XLM-R (Conneau et al., 2020) and mBERT (Devlin et al., 2019), following prior multilingual PET work. For each setup, we run 10 trials with stratified 70/15/15 train/dev/test splits. Trial indices are aligned across conditions: run i uses the same euphemism split for every baseline and transfer setup, enabling paired comparisons. Auxiliary datasets are split independently within each run. Performance is reported as the mean macro F1 across trials.

Inputs are capped at 512 tokens; longer examples are removed rather than truncated to avoid deleting the target expression. Boundary markers are implemented as the literal string [PET_BOUNDARY]; no new vocabulary item is added. Across fine-tuning runs, we use learning rate 1×10^{-5} , batch size 16, a maximum of 20 epochs, and early stopping with patience 5 on development macro F1.

We report three kinds of conditions. The pre-trained probe baseline uses the frozen pretrained encoder. Target-task fine-tuning trains the encoder on euphemism detection, either with boundaries or without boundaries. Auxiliary transfer trains the encoder on a non-euphemism auxiliary task before probing on euphemism detection. Target-task fine-tuning is therefore not an auxiliary-transfer condition; it is included to show how much can be gained from in-task supervision under each input format.

5 Results

Auxiliary transfer is unreliable. Table 2 reports mean macro F1 for direct euphemism fine-tuning and auxiliary-transfer conditions. For XLM-R on EUPH-B, most auxiliary tasks reduce performance. TroFi-B is the largest observed auxiliary gain, but the improvement over baseline is small (+0.023) and not statistically significant. On Euph-N, no XLM-R auxiliary task exceeds the pretrained baseline. mBERT shows more positive transfer cases, especially with Magpie-B, TroFi-B and Sensitivity, but gains are format-dependent and don’t approach target-task fine-tuning.

Target-span marking is the clearest positive contrast. The strongest observed gains come from direct euphemism fine-tuning with target-span-marked input. On EUPH-B, direct target fine-tuning with boundaries improves XLM-R from 0.655 to 0.755 and mBERT from 0.572 to 0.670. No auxiliary transfer condition approaches these gains.

Condition	EUPH-B		EUPH-NoB	
	XLm-R	mBERT	XLm-R	mBERT
Pretrained	0.655	0.572	0.645	0.580
<i>Target-task FT</i>				
B	0.755***	0.670**	0.653	0.609
NoB	0.659	0.621*	0.675*	0.617*
<i>Auxiliary transfer</i>				
Movie	0.633	0.555	0.604**	0.563
Books	0.624*	0.589	0.615*	0.581
Polite	0.630*	0.547	0.614***	0.543**
IDEM	0.624**	0.547	0.609**	0.549*
Magpie (N)	0.639	0.578	0.625*	0.558
Magpie (B)	0.668	0.635***	0.618*	0.579
Sensitive	0.580***	0.611**	0.606***	0.606*
TroFi (N)	0.629	0.592	0.612*	0.586
TroFi (B)	0.678	0.623***	0.612**	0.575
Multi	0.626*	0.589	0.603***	0.578

Table 2: Mean macro F1 over 10 trials. N = no boundary markers; B = boundary markers (classification tuned). Bolded are the highest scores in each section. Paired two-tailed t-test (h_0 : fine-tuned = pre), * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

	X-B	m-B	X-nB	m-nB
Euph	***	***	*	-
Trofi	**	*	-	-
Magpie	-	**	-	-

Table 3: Paired two-tailed t-tests (h_0 : train-B = train-nB) between the boundaried version and non-boundaried version of each training set (rows) and testing conditions (columns). X = XLmR, m = mBERT. B = euphB testing, nB = euph-noB testing. - $p > 0.05$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$. Highest score in each section bolded.

Boundary format matters, but not by itself.

Boundary markers are most useful when they are part of target-task fine-tuning and evaluation, and when the boundary settings of the training and testing set match. Comparing Magpie-B with Magpie-NoB on EUPH-B also suggests that boundary-marked auxiliary training is more useful when the target is boundary-marked. The same pattern holds for TroFi. We interpret this conservatively: transfer is associated with auxiliary tasks that combine contextual disambiguation with explicit target localization, not with figurative-language exposure alone.

Negative-transfer summary. Table 5 counts auxiliary conditions as positive, near-baseline, or negative using a ± 0.005 margin around the pretrained baseline. These counts are descriptive rather than significance tests (though paired T-test results are reported in tables 2 and 3). The pattern is clearest for XLm-R: auxiliary fine-tuning usu-

Condition	EUPH-B		EUPH-NoB	
	XLm-R	mBERT	XLm-R	mBERT
Pretrained	0.036	0.035	0.028	0.026
<i>Target-task FT</i>				
B	0.028	0.044	0.035	0.043
NoB	0.037	0.026	0.034	0.036
<i>Auxiliary transfer</i>				
Movie	0.040	0.033	0.043	0.031
Books	0.030	0.032	0.024	0.031
Polite	0.030	0.031	0.026	0.033
IDEM	0.040	0.032	0.036	0.018
Magpie (N)	0.030	0.033	0.020	0.031
Magpie (B)	0.021	0.040	0.022	0.036
Sensitive	0.027	0.025	0.016	0.029
TroFi (N)	0.032	0.027	0.037	0.026
TroFi (B)	0.040	0.029	0.038	0.037
Multi	0.027	0.025	0.035	0.022

Table 4: Standard deviation over 10 trials. N = no boundary markers; B = boundary markers (classification tuned). Bolded are the highest scores in each section.

Target	Model	Pos.	Near	Neg.
EUPH-B	XLm-R	2	0	8
EUPH-B	mBERT	7	0	3
EUPH-NoB	XLm-R	0	0	10
EUPH-NoB	mBERT	1	5	4

Table 5: Auxiliary-transfer outcomes relative to the pretrained probe. “Near” means within ± 0.005 macro F1 of baseline.

ally hurts, and removing target boundaries often resulted in significantly different scores that were lower than the pretrained value. mBERT is less brittle, but its improvements are not stable across target formats.

6 Discussion

The central assumption tested here is that pragmatic relatedness should be sufficient for useful transfer. Politeness, sentiment, figurative language and sensitivity are all reasonable auxiliary choices for euphemism detection, but they mostly do not improve performance and often reduce it. This matters because the choice is linguistically intuitive: euphemisms are pragmatic, so pragmatic supervision should help. Our results suggest that this assumption is unreliable in a PET-style setup.

One reason is that sentence-level pragmatic tasks and PET classification ask different questions. A euphemistic expression can change perceived politeness, sensitivity, or sentiment when compared with a more direct alternative. However, the PET data do not ask the model to compare a euphemistic phrasing with a direct rephrasing of the same meaning. Instead, they ask whether a given expression

is euphemistic in its current context. Global properties such as politeness or sentiment therefore need not systematically separate the classes.

The results are consistent with a structural-alignment explanation, but they do not isolate structure as the only causal factor. Magpie and TroFi differ from the pragmatic proxy tasks not only in boundary marking but also in literal/nonliteral label structure and data domain. We therefore interpret the pattern conservatively: auxiliary tasks that resemble PET-style euphemism detection in target localization and contextual disambiguation appear more promising than broad sentence-level pragmatic proxies in this setup.

More broadly, the results suggest that selecting auxiliary tasks based solely on linguistic intuition can be misleading. Euphemisms are widely discussed as pragmatic phenomena, making politeness, sentiment, and sensitivity appear to be natural transfer sources. However, transfer performance appears to depend not only on the linguistic phenomenon involved but also on whether source and target tasks require similar decisions about similar units of text.

In this work, we sought to understand whether pragmatic supervision helps language models recognize euphemistic use. The results suggest that task design does more than affect measured performance: it can shape what information a model is encouraged to represent and which transfer signals become useful. This matters when computational results are interpreted as evidence about pragmatic understanding. Human euphemism use often involves choosing or interpreting an expression relative to a more direct alternative and inferring its social or pragmatic effect in discourse. PET-style classification, by contrast, asks whether a particular expression is euphemistic in its current context. To compare model behavior with assumptions about human pragmatic perception, the task design should therefore align with the relevant natural-language use case: the decision unit, available span information, label semantics, and contrast set should match the inference being claimed. Preserving target-span information is one practical step, but the broader recommendation is to design transfer experiments around the intended pragmatic inference rather than choosing auxiliary data only by broad linguistic theme.

Limitations

Our experiments focus on English PETs and two multilingual encoders; transfer patterns may differ in other languages, with English-only encoders, or with instruction-tuned LLMs (see Appendix B). Boundary tokens are an artificial annotation mechanism, and we use only one marker string. Future work should test neutral marker pairs, character offsets, span pooling, and architectures explicitly conditioned on spans. We also do not evaluate sequential fine-tuning where the auxiliary-tuned model is further fine-tuned end-to-end on euphemism detection; such a setting could yield different behavior. To address points made in the Discussion Section, future work can also create paraphrasing for the PETs.

Finally, our design does not fully disentangle target localization from other forms of task similarity. Boundary-marked Magpie and TroFi differ from pragmatic proxy tasks in several ways, including literal/nonliteral labels, target-span structure, and data domain. We therefore treat structural alignment as a hypothesis supported by the observed pattern, not as an isolated causal mechanism.

7 AI Use

ChatGPT (OpenAI, 2026) was consulted for code and instruction tuning advice. All final decisions were made by a human.

Acknowledgments

Thank you to Anand Gopeekrishnan for providing advice and helping locate datasets.

This material is based upon work supported by the National Science Foundation under Grant Nos. 226006 and 2428506. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

- Julia Birke and Anoop Sarkar. 2007. Active learning for the identification of nonliteral language. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*, pages 21–28. Association for Computational Linguistics.
- Hasan Biyik, Patrick Lee, and Anna Feldman. 2024. Turkish delights: a dataset on Turkish euphemisms. In *Proceedings of the First Workshop on Natural Language Processing for Turkic Languages (SIGTURK)*.

- 2024), pages 71–80. Association for Computational Linguistics.
- John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447. Association for Computational Linguistics.
- Qihong Chen, Jiawei Li, Hyunjae Suh, Lianghao Jiang, Zheng Zhou, Jingze Chen, Jiri Gesi, and Iftekhar Ahmed. 2024. Does the order of fine-tuning matter and why? *arXiv preprint arXiv:2410.02915*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). *Preprint*, arXiv:1911.02116.
- Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 250–259. Association for Computational Linguistics.
- Verna Dankers, Christopher Lucas, and Ivan Titov. 2022. [Can transformer be too compositional? analysing idiom processing in neural machine translation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3608–3626, Dublin, Ireland. Association for Computational Linguistics.
- Mark Davies and Robert Fuchs. 2015. The glowbe corpus (global web-based english): A 1.9 billion word corpus of english from 20 countries. *English Corpus Linguistics: Crossing Paths*, pages 159–182.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Christian Felt and Ellen Riloff. 2020. Recognizing euphemisms and dysphemisms using sentiment analysis. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 136–145. Association for Computational Linguistics.
- Todd Firsich and Anthony Rios. 2024. Can GPT4 detect euphemisms across multiple languages? In *Proceedings of the 4th Workshop on Figurative Language Processing (FigLang 2024)*, pages 65–72. Association for Computational Linguistics.
- Martha Gavidia, Patrick Lee, Anna Feldman, and Jing Peng. 2022. CATs are fuzzy PETs: A corpus and analysis of potentially euphemistic terms. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2658–2671. European Language Resources Association.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Gilles Grolleau, Naoufel Mzoughi, Deborah Peterson, and Marjorie Tendero. 2022. Changing the world with words? euphemisms in climate change issues. *Ecological Economics*, 193:107307.
- Hessel Haagsma, Johan Bos, and Malvina Nissim. 2020. MAGPIE: A large corpus of potentially idiomatic expressions. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 279–287. European Language Resources Association.
- Daniel Han, Michael Han, and Unsloth team. 2026. [Unsloth](#).
- Patrick Lee, Alain Chirino Trujillo, Diana Cuevas Plancarte, Olumide Ojo, Xinyi Liu, Iyanuoluwa Shode, Yuan Zhao, Anna Feldman, and Jing Peng. 2024. MEDs for PETs: Multilingual euphemism disambiguation for potentially euphemistic terms. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 875–881. Association for Computational Linguistics.
- Patrick Lee and Anna Feldman. 2024. [Report on the multilingual euphemism detection task](#). In *Proceedings of the 4th Workshop on Figurative Language Processing (FigLang 2024)*, pages 110–114, Mexico City, Mexico (Hybrid). Association for Computational Linguistics.
- Patrick Lee, Martha Gavidia, Anna Feldman, and Jing Peng. 2022. Searching for PETs: Using distributional and sentiment-based methods to find potentially euphemistic terms. In *Proceedings of the Second Workshop on Understanding Implicit and Underspecified Language*, pages 22–32. Association for Computational Linguistics.
- Patrick Lee, Iyanuoluwa Shode, Alain Trujillo, Yuan Zhao, Olumide Ojo, Diana Plancarte, Anna Feldman, and Jing Peng. 2023. FEED PETs: Further experimentation and expansion on the disambiguation of potentially euphemistic terms. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 437–448. Association for Computational Linguistics.
- Yelena Mejova, Amy X. Zhang, Nicholas Diakopoulos, and Carlos Castillo. 2014. [Controversy and sentiment in online news](#). *Preprint*, arXiv:1409.8152.

OpenAI. 2026. [Chatgpt](#).

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124.

Whitney Poh, Julia Sammartino, Jasper Andrew, Witold Kieraś, Natalia Zawadzka-Paluckta, Iryna Dilai, Libby Barak, Jing Peng, and Anna Feldman. 2026. [The multilingual euphemism benchmark: Datasets and baselines for pragmatic language understanding](#). In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 7307–7319, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Alexander Prochnow, Johannes E. Bendler, Caroline Lange, Foivos Ioannis Tzavellos, Bas Marco Göritzer, Marijn ten Thij, and Riza Batista-Navarro. 2024. IDEM: The Idioms with EMotions dataset for emotion recognition. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8569–8579. ELRA and ICCL.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordani, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. [Exploring and predicting transferability across nlp tasks](#). *Preprint*, arXiv:2005.00770.

Yuting Wang, Yiyi Liu, Ruqing Zhang, Yixing Fan, and Jiafeng Guo. 2022. Euphemism Detection by Transformers and Relational Graph Attention Network. In *Proceedings of the 3rd Workshop on Figurative Language Processing*, pages 79–83. Association for Computational Linguistics.

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. 2019. [Characterizing and avoiding negative transfer](#). *Preprint*, arXiv:1811.09751.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

Wanzheng Zhu, Hongyu Gong, Rohan Bansal, Zachary Weinberg, Nicolas Christin, Giulia Fanti, and Suma Bhat. 2021. [Self-supervised euphemism detection](#)

and identification for content moderation. *CoRR*, abs/2103.16808.

A Dataset Labeling

A.1 Euphemisms

Euphemisms (Lee et al., 2024; Lee and Feldman, 2024) dataset can be found at <https://github.com/pl464/euph-detection-datasets>

A.2 Magpie

Magpie (Haagsma et al., 2020; Dankers et al., 2022) retrieved from <https://huggingface.co/datasets/gstarti/magpie>

A.3 Trofi

Trofi (Birke and Sarkar, 2007) retrieved from <https://huggingface.co/datasets/tasksource/TroFi>

A.4 IDEM

IDEM (Prochnow et al., 2024) dataset can be found at https://github.com/AlexanderProchnow/idem/blob/main/dataset/idem_train.csv

A.5 Sentiment

Sentiment (Pang and Lee, 2005) retrieved from <https://www.cs.cornell.edu/people/pabo/movie-review-data/rt-polaritydata.tar.gz>

A.6 Sensitivity

Following Lee et al. (2023)’s methods, we use a binary sensitive vs. non-sensitive dataset labeled via semantic similarity to a seed lexicon of sensitive/taboo topics³(Mejova et al., 2014). **Leakage control:** because both Sensitivity and PET data are drawn from web corpora GloWbE (Davies and Fuchs, 2015), we remove any Sensitivity instances that overlap with PET instances to avoid cross-task contamination.

A.7 Politeness

The original labels from the Stanford Politeness Dataset (Danescu-Niculescu-Mizil et al., 2013) have been converted to binary labels, with negative labels being 0 and positive labels being 1. Retrieved from https://github.com/Genius1237/TyDiP/blob/main/data/all/en_train.csv

³<https://drive.google.com/file/d/19PyIVklLtUgfunSFxgVvMAY5z6gjaAwT/view>

A.8 Books vs Electronics

The dataset from which we drew our Books vs Electronics data, released by [Blitzer et al. \(2007\)](#), is a sentiment dataset, however, we took the books and electronics reviews and labeled them according to topic (book review or electronics review) instead of using the sentiment information. Retrieved from <https://www.cs.jhu.edu/~mdredze/datasets/sentiment/index2.html>

B Exploratory Extension to Instruction-Tuned Models

B.1 Setup

In addition, we use LoRA to fine-tune Qwen3-4B ([Yang et al., 2025](#)) and Llama-3.1-8B-Instruct ([Grattafiori et al., 2024](#)) models. Parameters are as follows:

LoRA settings were:

- Targeted modules: q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj
- r: 16
- alpha: 16

Training settings were:

- Maximum training sequence length: 4096
- Learning rate: 1e-5
- Epochs: 2

Prompts for this could be found in Appendices B.2 and B.3. These models were fine-tuned using Unsloth ([Han et al., 2026](#)) and Google Colab’s G4 GPU⁴. When G4’s were not available, A100s were used.

Because instruction-tuning on auxiliary tests may lead to overfitting, we also ran an additional trial where euphemism training data was added to the instruction-tuning data. The euphemism training data mixed into the instruction-tuning data includes boundary and non-boundary data with both boundary and non-boundary prompts (see the Appendix for more information). This setting uses 70% task data, 10% euphemism with boundaries (boundary prompt), 10% euphemism with boundaries (no boundary prompt) and 10% euphemism without boundaries for training, with the normal

euphemism test data, and is referred to as Qwen-Mix or Llama-Mix in the results tables. The same example may or may not be present in the training data for each split with both boundaries and without, however, the training data and the test data are separate and do not overlap.

The instruction-tuned models do not use the same training/validation/testing splits as the classification-tuned transformer models. However, the splits for each dataset are the same for all instruction-tuned models.

B.2 System Prompt

You are a linguist who studies pragmatic language. You understand things like euphemisms, politeness, metaphors, idioms and topic sensitivity. You use that knowledge to complete the task required and output the appropriate response.

B.3 Content Prompt

- Euphemisms (no boundaries): Can you decide whether the text contains a euphemism or not? Output 'Output: 1' if it contains a euphemism, 'Output: 0' if not. Do not include an intro or conclusion, simply output the number. Here is the text:
- Euphemisms (boundaries): Can you decide whether the text between [PET_BOUNDARY] markers is used euphemistically or not? Output 'Output: 1' if it's euphemistic, 'Output: 0' if not. Do not include an intro or conclusion, simply output the number. Here is the text:
- Idioms (no boundaries): Can you decide whether the given text contains an idiom or not? Output 'Output: 1' if there is an idiom, 'Output: 0' if not. Do not include an intro or conclusion, simply output the number. Here is the text:
- Idioms (boundaries): Can you decide whether the word inside the [PET_BOUNDARY] markers within the given text is used idiomatically or not? Output 'Output: 1' if idiomatic, 'Output: 0' if not. Do not include an intro or conclusion, simply output the number. Here is the text:
- Sensitivity: Can you decide whether this text is about a sensitive topic or not? Output 'Output: 1' if sensitive, 'Output: 0' if not. Do not

⁴<https://colab.research.google.com>

include an intro or conclusion, simply output the number. Here is the text:

- Politeness: Can you decide whether the content of the given text is polite or not? Output 'Output: 1' if polite, 'Output: 0' if impolite. Do not include an intro or conclusion, simply output the number. Here is the text:
- Metaphors (boundaries): Can you decide whether the word inside the [PET_BOUNDARY] markers within the given text is used metaphorically or not? Output 'Output: 1' if metaphorical, 'Output: 0' if not. Do not include an intro or conclusion, simply output the number. Here is the text:
- Metaphors (no boundaries): Can you decide whether the given text contains a metaphor or not? Output 'Output: 1' if there is a metaphor, 'Output: 0' if not. Do not include an intro or conclusion, simply output the number. Here is the text:
- Sentiment (movies): Can you decide whether this movie review is positive or negative in terms of sentiment? Output 'Output: 1' if positive, 'Output: 0' if negative. Do not include an intro or conclusion, simply output the number. Here is the text:
- Sentiment (idioms): Can you decide whether this text, which contains an idiom, is of positive or negative sentiment? Output 'Output: 1' if it's positive, 'Output: 0' if it's negative. Do not include an intro or conclusion, simply output the number. Here is the text:
- Books: Can you decide whether this text is a book review or electronics review? Output 'Output: 1' if it's a book review, 'Output: 0' if it's an electronics review. Do not include an intro or conclusion, simply output the number. Here is the text:

Our instruction-tuned models performed poorly at euphemism detection. Table 6 shows results for the instruction-tuned LLMs. Previous work (Poh et al., 2026) have demonstrated that smaller LLMs did not always have great understanding of euphemisms and our results were consistent with that. Models tend to overpredict one class when

Condition	EUPH-B				EUPH-NoB			
	L	LMix	Q	QMix	L	LMix	Q	QMix
Pretrained	0.408	-	0.54	-	0.543	-	0.55	-
<i>Target-task FT</i>								
B	0.601	-	0.605	-	0.450	-	0.414	-
B2	0.544	-	0.604	-	0.484	-	0.379	-
NoB	0.436	-	0.362	-	0.580	-	0.573	-
<i>Auxiliary transfer</i>								
Movie	0.433	0.501	0.512	0.557	0.587	0.570	0.559	0.511
Books	0.506	0.507	0.520	0.560	0.553	0.550	0.520	0.460
Polite	0.511	0.515	0.474	0.556	0.547	0.546	0.548	0.502
IDEM	0.514	0.514	0.542	0.568	0.598	0.579	0.562	0.510
Magpie (N)	0.555	0.540	0.460	0.540	0.579	0.554	0.570	0.517
Magpie (B)	0.572	0.550	0.551	0.577	0.564	0.552	0.483	0.469
Magpie (B2)	0.593	0.561	0.510	0.590	0.516	0.528	0.518	0.464
Sensitive	0.406	0.435	0.449	0.530	0.586	0.586	0.518	0.745
TroFi (N)	0.515	0.512	0.355	0.466	0.587	0.580	0.555	0.563
TroFi (B)	0.576	0.563	0.506	0.543	0.550	0.543	0.524	0.500
TroFi (B2)	0.528	0.551	0.518	0.531	0.554	0.531	0.505	0.459

Table 6: Mean macro F1 over 10 trials. N = no boundary markers; B = boundary markers; B2 = data has boundaries but prompt is for non-boundaried data. B2 refers to the trials with the boundaried data but non-boundaried prompts (instruction-tuned). Q = Qwen3-4B, L = Llama-3.1-8B-Instruct

prompted to identify euphemisms. This is documented in table 7. Only the target task instruction-tuned models with the correct boundary setting were consistently not skewed towards one class. Because of these highly imbalanced results, we do not discuss instruction tuned models in detail in other sections of this paper. However, note that changing the boundary labeling information could completely flip class decisions. Several tasks were also tested on their corresponding test splits to see whether the models only struggled with euphemisms or whether they struggled with other tasks as well, and these results are reported in table 9. The results show F1s generally above 0.7 with low output class imbalance, indicating that smaller LLMs may have specific problems with euphemism detection. The tasks with F1s lower than 0.7 were all idiom and metaphor tasks, indicating that figurative language may be challenging for language models.

C Additional Tables

Tables 7 and 9 show that there is high class imbalance in the outputs when a model is fine-tuned on one task and then tested on another. As demonstrated by table 10, using the same prompt in training and testing reduces class imbalance, though it does not increase F1. These results show that a model instruction-tuned on one task pretty much only learns that task, but does not retain that information and utilize it to aid it in completing a different task with a different prompt. Even in

	Q	Q-B	L	L-B
Pre	0.30	0.78	0.76	0.95
Trofi	0.72	0.99	0.43	0.84
Trofi-B	0.28	0.83	0.29	0.76
Trofi-B2	0.23	0.94	0.30	0.83
Magpie	0.38	0.90	0.37	0.79
Magpie-B	0.17	0.69	0.32	0.77
Magpie-B2	0.24	0.89	0.21	0.69
Euph	0.50	0.98	0.49	0.93
Euph-B	0.06	0.50	0.10	0.69
Euph-B2	0.04	0.55	0.15	0.80
Movie	0.34	0.83	0.61	0.93
Polite	0.31	0.88	0.29	0.85
Sens	0.24	0.92	0.63	0.95
Books	0.25	0.81	0.32	0.84
Idem	0.34	0.80	0.50	0.84

Table 7: Average percent of outputs predicted as positive class (test euphemisms). Q = Qwen, L = Llama, B = Boundaried test data. N = 10 trials.

the setups where the supplementary task is mixed with euphemism training data of different boundary label styles, the F1s do not increase above the models instruction-tuned only on euphemisms. However, the class imbalance can be reduced by task-mixing, especially for the euphemism testing task with boundaries, though a rather heavy level of skew still remains. In both cases, the model tends to overpredict 0 for euphemism detection without boundaries, and 1 for euphemism detection with boundaries.

Table 11 shows that this is not true for classification-tuned transformers models, as while some results may be mediocre, models that are fine-tuned on one dataset and tested on another are generally not skewed towards one class. This is because we fitted a separate logistic regression model using the encoder to encode the testing task (euphemisms) instead of directly using the classifier from the training task. However, we provide the table anyways to show that the testing data is split randomly from the training data and therefore isn't skewed towards one class.

D Examples of Tasks and Data

Table 12 shows examples of the auxiliary tasks. Table 13 show examples of Examples of politeness, idiomaticity and sentiment in the euphemism data.

	QM	QM-B	LM	LM-B
Trofi	0.39	0.89	0.40	0.86
Trofi-B	0.19	0.75	0.29	0.79
Trofi-B2	0.13	0.78	0.27	0.76
Magpie	0.23	0.74	0.30	0.81
Magpie-B	0.14	0.60	0.31	0.80
Magpie-B2	0.13	0.62	0.23	0.75
Movie	0.24	0.69	0.36	0.86
Polite	0.20	0.69	0.27	0.84
Sens	0.14	0.78	0.47	0.92
Books	0.12	0.69	0.30	0.86
Idem	0.20	0.68	0.37	0.83

Table 8: Average percent of outputs predicted as positive class (test euphemisms). QM = QwenMix, L = LlamaMix, B = Boundaried test data. N = 10 trials.

	Q-F1	Q-Imba	L-F1	L-Imba
Trofi	0.58	0.49	0.59	0.50
Trofi-B	0.65	0.52	0.66	0.44
Trofi-B2	0.67	0.54	0.59	0.69
Magpie	0.73	0.52	0.75	0.55
Magpie-B	0.71	0.47	0.61	0.80
Magpie-B2	0.75	0.55	0.70	0.69
Movie	0.92	0.49	0.93	0.47
Polite	0.71	0.43	0.77	0.55
Sens	0.71	0.50	0.74	0.52
Books	0.98	0.50	0.98	0.50
Idem	0.95	0.50	0.95	0.50

Table 9: Macro F1 and average percent of outputs predicted as positive class (test orig. task). Q = Qwen, L = Llama, B = Boundaried test data. N = 10 trials.

	F1	Imba
TrofiB	0.558	0.62

Table 10: A Qwen3-4B model instruction-tuned on the TrofiB data, with the EuphB prompt. Results show Macro F1 and average percent of outputs predicted as positive class over 10 trials.

	mBert	mBERT-B	XLMR	XLMR-B
Trofi	0.50	0.50	0.52	0.51
Trofi-B	0.51	0.50	0.51	0.50
Magpie	0.50	0.49	0.51	0.51
Magpie-B	0.49	0.51	0.50	0.50
Movie	0.50	0.51	0.50	0.50
Polite	0.51	0.51	0.50	0.51
Sens	0.51	0.51	0.50	0.50
Books	0.50	0.49	0.52	0.51
Idem	0.52	0.51	0.52	0.51

Table 11: Average percent of outputs predicted as positive class (test euphemisms) for classification-tuned transformer models. N = 10 trials.

D.1 Results on the no-boundary target (Euph-NoB)

We provide supplementary tables with the results on the no-boundary target. Table 14 shows the results using XLM-R and Table 15 shows the results using mBERT.

D.2 Per-category results (Euph-B)

The paper discusses the overall results for the data. The euphemism data set includes category annotation for each euphemistic expression including categories such as politics and death. Tables 16 and 17 shows the breakdown of the results for each semantic category for XLM-R and mBERT respectively.

³<https://huggingface.co/datasets/gsarti/magpie>

⁴<https://huggingface.co/datasets/tasksource/Trofi>

⁵<https://www.cs.cornell.edu/people/pabo/movie-review-data/rt-polaritydata.tar.gz>

⁶https://github.com/Genius1237/TyDiP/blob/main/data/all/en_train.csv

⁷Indices from https://github.com/pl464/euph-detection-datasets/blob/main/FigLang_2024/EN_PET_dataset_2024.csv

Dataset	Source and Source URL	Text	Label
Magpie	Haagsma et al. (2020); Dankers et al. (2022) ³	“[PET_BOUNDARY] Spotted on [PET_BOUNDARY] a fan in a department store.”	0
Magpie	Haagsma et al. (2020) ³	“Everest appeared [PET_BOUNDARY] out of the question, [PET_BOUNDARY] such was his temperamental and erratic nature.”	1
Trofi	Birke and Sarkar (2007) ⁴	“And most of the jobs will go to candidates sought out by search firms , not those knocking on headhunter doors .”	0
Trofi	Birke and Sarkar (2007) ⁴	“That planted the seeds of discontent between Mr. Kirk and the Jackson forces .”	1
Sentiment	Pang and Lee (2005) ⁵	“obviously , a lot of people wasted a lot of their time (including mine) on something very inconsequential .”	0
Sentiment	Pang and Lee (2005) ⁵	“more romantic , more emotional and ultimately more satisfying than the teary-eyed original .”	1
Sensitivity	Lee et al. (2023); Davies and Fuchs (2015); Mejova et al. (2014)	“It was a nice try , my friend . Bravo . Well played . However , you were pwned before you posted .”	0
Sensitivity	Lee et al. (2023); Davies and Fuchs (2015); Mejova et al. (2014)	“So , when it lacks a budget for direction , Congress still has the power to spend money and keep government operating , but the process can get mucked up .”	1
Politeness	Danescu-Niculescu-Mizil et al. (2013) ⁶	“Don’t talk about me, in my virtual presence, and tell someone else to ignore me. Do you understand why this behavior is inappropriate?”	0
Politeness	Danescu-Niculescu-Mizil et al. (2013) ⁶	“Good job working on the list, thanks. Could you make sure you move them from "To do" to "Done" when you finish?”	1

Table 12: Text examples from the auxiliary datasets. Labels represent the labels we use for our fine-tuning task, not necessarily the provided labels in the original creators’ dataset.

Type	Text	Orig. Index ⁷
Polite	“Dear Michael, Thank you for your pipcture and for sharing wit with the world. My own son is 11 now and I recall seeing your lovely picture when I was [PET_BOUNDARY]expecting[PET_BOUNDARY] him and I felt a strong connection to it. Thank you also for your courageous and continuous fight for the truth.”	657
Impolite	“I don’t condone violence but had she butted out like Adriana asked her to and not followed Adriana looking for a fight, she might not have gotten hit at all!! I really think she has a [PET_BOUNDARY]drinking problem[PET_BOUNDARY]. She needs help!!!”	550
Positive Sentiment	“In spite of his [PET_BOUNDARY]advanced age[PET_BOUNDARY], Rollins remains one of jazz’s most talented improvisers.”	73
Negative Sentiment	“This speaker should be disinvited from further commencements and if he is employed by ANY scholl , any level be [PET_BOUNDARY]dismissed[PET_BOUNDARY]”	518
Idiomatic	“There are plenty of people who say the show is kind of [PET_BOUNDARY]over the hill[PET_BOUNDARY]”	1197
Not Idiomatic	“Early Sunday, the Heath Ministry confirmed that the novel coronavirus claimed 19 [PET_BOUNDARY]fatalities[PET_BOUNDARY]”	723

Table 13: Examples from Lee and Feldman (2024) that contain certain pragmatic traits. Long examples may be truncated. Pragmatic categorizations were determined by the authors of this paper.

XLM-R (Euph-NoB)	Phys/Ment	Misc	Body	Subs	Pol	Emp/Fin	Sex	Death
Baseline	0.674	0.513	0.693	0.742	0.506	0.556	0.675	0.656
Euph fine-tune (B→NoB)	0.656	0.543	0.693	0.675	0.519	0.630	0.646	0.674
Euph fine-tune (NoB→NoB)	0.689	0.536	0.709	0.708	0.534	0.633	0.688	0.702
TroFi (NoB)	0.626	0.552	0.660	0.647	0.577	0.531	0.615	0.634
TroFi (B)	0.614	0.540	0.695	0.622	0.569	0.575	0.595	0.625
Movie Sentiment	0.600	0.465	0.638	0.651	0.599	0.580	0.623	0.603
Books vs Electronics	0.609	0.511	0.674	0.684	0.561	0.533	0.635	0.655
Magpie (NoB)	0.643	0.536	0.621	0.697	0.517	0.582	0.625	0.625
Magpie (B)	0.632	0.507	0.710	0.616	0.510	0.583	0.617	0.606
Sensitivity	0.590	0.545	0.694	0.659	0.540	0.576	0.595	0.589
Politeness	0.629	0.521	0.654	0.630	0.565	0.570	0.575	0.635
Idiom Sentiment	0.627	0.526	0.649	0.669	0.492	0.534	0.625	0.601
Multi	0.610	0.595	0.590	0.636	0.521	0.558	0.573	0.595

Table 14: Per-category macro F1 for XLM-R when the euphemism target has no boundary markers (Euph-NoB).

mBERT (Euph-NoB)	Phys/Ment	Misc	Body	Subs	Pol	Emp/Fin	Sex	Death
Baseline	0.590	0.505	0.662	0.588	0.438	0.550	0.613	0.555
Euph fine-tune (B→NoB)	0.599	0.502	0.618	0.611	0.430	0.597	0.650	0.625
Euph fine-tune (NoB→NoB)	0.625	0.504	0.627	0.637	0.421	0.592	0.620	0.625
TroFi (NoB)	0.609	0.496	0.623	0.614	0.464	0.577	0.626	0.545
TroFi (B)	0.570	0.489	0.642	0.604	0.397	0.546	0.589	0.587
Movie Sentiment	0.580	0.499	0.611	0.587	0.460	0.545	0.555	0.531
Books vs Electronics	0.589	0.528	0.644	0.581	0.473	0.534	0.581	0.589
Magpie (NoB)	0.573	0.495	0.579	0.567	0.373	0.550	0.584	0.539
Magpie (B)	0.602	0.527	0.576	0.648	0.437	0.527	0.582	0.567
Sensitivity	0.603	0.503	0.669	0.643	0.504	0.565	0.607	0.626
Politeness	0.550	0.479	0.586	0.559	0.526	0.505	0.567	0.521
Idiom Sentiment	0.556	0.519	0.602	0.512	0.424	0.546	0.550	0.521
Multi	0.573	0.583	0.565	0.611	0.408	0.530	0.586	0.581

Table 15: Per-category macro F1 for mBERT when the euphemism target has no boundary markers (Euph-NoB).

XLM-R (Euph-B)	Phys/Ment	Misc	Body	Subs	Pol	Emp/Fin	Sex	Death
Baseline	0.671	0.629	0.708	0.696	0.474	0.584	0.661	0.654
Euph fine-tune (B→B)	0.741	0.717	0.704	0.833	0.663	0.748	0.722	0.757
Euph fine-tune (NoB→B)	0.683	0.535	0.646	0.675	0.568	0.633	0.672	0.660
TroFi (NoB)	0.621	0.584	0.670	0.684	0.588	0.571	0.632	0.648
TroFi (B)	0.690	0.559	0.750	0.675	0.549	0.636	0.641	0.717
Movie Sentiment	0.632	0.537	0.622	0.649	0.665	0.616	0.655	0.619
Books vs Electronics	0.629	0.566	0.655	0.694	0.556	0.545	0.636	0.634
Magpie (NoB)	0.672	0.526	0.653	0.637	0.546	0.600	0.666	0.633
Magpie (B)	0.665	0.571	0.722	0.759	0.490	0.630	0.653	0.678
Sensitivity	0.572	0.554	0.622	0.608	0.479	0.555	0.573	0.580
Politeness	0.638	0.569	0.675	0.646	0.501	0.601	0.601	0.641
Idiom Sentiment	0.651	0.559	0.658	0.670	0.500	0.562	0.588	0.620
Multi	0.623	0.581	0.632	0.709	0.512	0.577	0.652	0.625

Table 16: Per-category macro F1 for XLM-R when the euphemism target includes boundary markers (Euph-B).

mBERT (Euph-B)	Phys/Ment	Misc	Body func./part	Subs	Politics	Empl/Fin	Sex	Death
Baseline	0.571	0.507	0.631	0.589	0.542	0.535	0.592	0.559
Euph fine-tune (B→B)	0.673	0.593	0.641	0.691	0.533	0.689	0.652	0.654
Euph fine-tune (NoB→B)	0.630	0.540	0.669	0.665	0.439	0.596	0.587	0.619
TroFi (NoB)	0.598	0.480	0.675	0.597	0.493	0.571	0.659	0.566
TroFi (B)	0.633	0.591	0.635	0.658	0.509	0.579	0.649	0.618
Movie Sentiment	0.585	0.480	0.598	0.505	0.447	0.531	0.565	0.539
Books vs Electronics	0.572	0.501	0.613	0.627	0.480	0.548	0.611	0.620
Magpie (NoB)	0.582	0.510	0.590	0.616	0.429	0.552	0.560	0.597
Magpie (B)	0.645	0.558	0.699	0.713	0.480	0.599	0.588	0.646
Sensitivity	0.609	0.487	0.682	0.657	0.446	0.571	0.639	0.626
Politeness	0.542	0.466	0.639	0.527	0.547	0.487	0.617	0.536
Idiom Sentiment	0.551	0.502	0.456	0.542	0.472	0.524	0.542	0.544
Multi	0.548	0.546	0.607	0.635	0.503	0.546	0.600	0.628

Table 17: Per-category macro F1 for mBERT when the euphemism target includes boundary markers (Euph-B).

Risk-Averse Failure Modes of Jensen-Shannon Distance in RLVR Distributional Alignment

Sebastian Loftus^{1*}, Xinpeng Wang^{2,3}, Marie-Catherine de Marneffe^{1,4}, Barbara Plank^{2,3}

¹CENTAL, UCLouvain, Belgium, ²CIS, LMU Munich, Germany,

³Munich Center for Machine Learning, Germany, ⁴FNRS, Belgium

sebastian.loftus@uclouvain.be, xinpeng@cis.lmu.de, marie-catherine.demarneffe@uclouvain.be,
b.plank@lmu.de

Abstract

Jensen-Shannon Distance (JSD) appears well-suited as a reward signal for aligning model predictions with human annotations, particularly for subjective tasks where label variation reflects legitimate disagreement. We demonstrate that JSD’s mathematical properties create exploitable reward landscapes in reinforcement learning. Via a study on preference distribution prediction, we show that two JSD-based reward formulations both fail to achieve alignment and instead lead to models converging at conservative, near-uniform predictions that maximize expected reward without matching human behavior. We trace this failure to JSD’s non-uniform sensitivity: it penalizes errors near balanced distributions less than errors near extreme distributions, creating risk asymmetry that incentivizes safe predictions over accurate ones. Our findings reveal fundamental challenges in using bounded symmetric divergence metrics as scalar rewards for distributional alignment.

1 Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) emerged as a powerful LLM post-training strategy, capable of eliciting complex reasoning behavior on structured tasks, like math or coding (Shao et al., 2024; DeepSeek-AI, 2025). For these types of tasks, verifiable reward functions allow for an objective evaluation of model outputs, rewarding sound reasoning behavior. Extending the method to non-deterministic tasks with an inherent degree of human label variation (HLV) (Plank, 2022) poses a larger challenge, as traditional verifiers fail at capturing the inherent model uncertainty of the predicted label, necessary to account for human variation. To quantify models’ capabilities to accurately represent a pluralistic label space, a common

approach is to measure the similarity of the two distributions using distance metrics. We investigate using the Jensen-Shannon distance (JSD) to formulate a scalar reward based on the divergence between human and model label distributions, which we in turn use to fine-tune LLMs via RLOO (Ahmadian et al., 2024) and on the HelpSteer2 preference prediction dataset (Wang et al., 2025a), where the task is to predict human preferences between two model-generated outputs. Rather than improving human alignment, the fine-tuned models exhibited a severe failure mode, collapsing into predicting purely balanced distributions. An analysis of the possible values JSD can take in binary tasks reveals that JSD’s non-uniform sensitivity makes it highly susceptible to reward hacking, rendering it an unsuitable reward estimator for this task. We detail which distance metric properties lead to the observed behavior and what this implies for distributional reward function design.

2 Related Work

An increasing body of work explores distributional rewards to capture subjective variation. Siththaranjan et al. (2024) frame preference learning as inherently distributional and model the reward as a distribution. Zhang et al. (2025) train a mean-variance reward model and optimize it using the KL-Divergence loss on preference data. Zawar et al. (2024) compare using KL-Divergence and JSD for reward modeling in Multi-Agent Constrained Policy Optimization and find that the latter offers a more robust alternative to KL, due to its symmetry. The most similar to our approach is Lu et al. (2026), who align LLM uncertainty to human annotator variation, wherein they define an uncertainty calibration reward based on the mean absolute error between model predictions and human annotator distributions and use GRPO for optimization. In our work, we show that for a similar approach,

*Work done primarily while the author was a Master’s student at LMU.

certain distance metrics, like JSD lead to a failure mode and detail what general properties in distance metrics lead to this behavior.

3 Reward Functions

This section presents the formulation of the two JSD-based reward functions.

3.1 JSD-based Reward Estimation

Jensen-Shannon Distance For measuring the similarity between the model-predicted distribution and human annotator judgments, we employ JSD, i.e., the bidirectional extension of the KL-divergence. An advantage of using the JS Distance compared to JS Divergence is the fact that it satisfies the requirements for being a metric, making it a better-scaled distance and thus better for interpreting the results (Endres and Schindelin, 2003). JSD is defined as:

$$\text{JSD} = \sqrt{\frac{1}{2}D_{\text{KL}}(H \parallel M) + \frac{1}{2}D_{\text{KL}}(M \parallel H)}$$

Reward Functions The following details the design of the JSD-based reward functions. For each function, the input is comprised of two probability distributions, M and H , over the possible labels, where M is the model-predicted distribution and H is the distribution of human-annotated labels.

For the **Direct Approach**, we define the reward as $R_{\text{direct}} = 1 - \text{JSD}(H \parallel M)$, so that predictions closely matching the human distribution yield rewards near 1, while more divergent predictions approach 0.

Next, the **Binned Approach** estimates the reward by placing $\text{JSD}(H \parallel M)$ into reward bins. By mapping JSD to specified rewards, we can provide larger reward signals for more accurate predictions, thereby rewarding exploitation. For the case that no parsable probability distribution is generated, the model is penalized with a score of -5 to encourage valid distributions. A detailed overview of the bins can be found in Table 1. Formally, this is defined as a piecewise function:

$$R_{\text{binned}} = \begin{cases} B(\text{JSD}), & \text{if output is valid} \\ -5, & \text{if output is unparsable} \end{cases}$$

where $B(\cdot)$ represents the mapping function that assigns the JSD score to its corresponding reward bin.

JSD	0.0	< 0.1	< 0.2	< 1.0	undefined
Reward	10	5	2	0	-5

Table 1: Bins for the Binned approach, where JSD is $\text{JSD}(H \parallel M)$.

4 Experiment

The following section details the experimental setup based on Chen et al. (2024)’s setup for employing LLM-as-a-judge for distribution prediction for NLI.

4.1 Setup

Data We use the HelpSteer2-Preference (Wang et al., 2025b) dataset, which contains 7,118 prompts and preference pairs. To avoid overfitting to the most common distribution in the training data in the fine-tuning experiments, we prune the dataset to keep a more balanced set of distributions. For each unique distribution, we keep up to 120 samples, yielding 666 in total. These were split 70/15/15 into 466 training samples and 100 each for testing and validation; see Appendix A.8 for an example.

Models We use two models from different model families for the experiments to account for variance in model architecture. In an effort to reduce computational costs and to keep the necessary compute requirements to a minimum, smaller-scale models were chosen. The two models Qwen2.5-3B-Instruct,¹ and Llama-3.2-3B-Instruct² are henceforth referred to as Qwen and Llama.

Algorithm We use the REINFORCE Leave-one-out (RLOO) algorithm for reward-based fine-tuning. RLOO treats each complete generation as a single action and optimizes the reward for the entire sequence. It reduces variance by generating k samples per prompt and computing each sample’s baseline as the average reward of the other $k-1$ samples. The policy gradient is estimated by averaging these k individual gradient estimates (Williams, 1992; Ahmadian et al., 2024; Kool et al., 2019).

Implementation To extract the model’s probability distribution, we employ the verbalized probability expression approach, where the prediction is extracted directly from the LLM’s generated output.

¹<https://huggingface.co/Qwen/Qwen2.5-3B-Instruct>

²<https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct>

Compared to other expression methods, verbalization has been found to be the closest calibrated to human judgment (Meister et al., 2025). The final distribution was extracted from the generation using regular expressions. The detailed prompt can be found in Appendix A.1. To calculate the JSD, we used the jensenshannon method of the scipy.entropy library³ with the base e, bounding the possible values in $[0, \sqrt{\ln 2}] \approx [0, 0.833]$. To determine whether the failure mode is unique to the verbalization approach, we do additional tests using a logit-based approach, but after fine-tuning the model’s behavior did not change significantly and as such, we focused on the verbalized approach (see Appendix A.7 for further details on the logit approach).

4.2 Evaluation Metrics

To evaluate the method, we utilize distance correlation (Székely et al., 2007) and average Jensen-Shannon Divergence (JSD) to assess overall model performance across the test set. Specifically, distance correlation captures global trends, while average JSD evaluates performance at the instance level. Additionally, we calculate average entropy to analyze the distributional shape of the predictions both before and after fine-tuning.

Dist.-Corr. captures global dependence between model and human predictions (0 = independent, 1 = perfectly dependent), detecting both linear and nonlinear relationships, and is computed across the test set.

Average JSD is computed per instance and averaged over the test set and shows whether training was able to minimize the objective, but does not give information on the actual divergence of the distributions.

Average Entropy of the predictions before and after fine-tuning is reported to gain an understanding of the general shape of the model-predicted distributions, where values close to 1.0 indicate balanced distributions and 0.0 indicate imbalanced distributions.

4.3 Results

The following results demonstrate that the models converged to output balanced predictions (see Table 3). We focus on the change of the predicted

distributions before and after fine-tuning, as this is the main point of interest for the later discussion.

Distributions Analyzing the mean entropy shows that across both models and reward functions, the entropy rises above 0.94, indicating highly balanced predictions. Notably, Llama-base initially already had high average entropy, and as such, the difference before and after fine-tuning is not as large. Averaging the predictions shows that both base models show a pronounced position bias toward Response A - approximately [0.6/0.4] for Qwen and [0.7/0.3] for Llama, see Figure 3 in Appendix A.5. This bias, a well-documented issue in LLM evaluation (Wang et al., 2024), does not affect our validation study, as our goal is to demonstrate mode collapse, which remains evident even under the bias. This bias persists after fine-tuning in Qwen-direct, Llama-direct, and Llama-bins ([0.6/0.4]) and in Qwen-bins ([0.4/0.6]), which might be attributed to the inherent position bias of both models.

Model	Version	Dist.-Corr.↑	Avg. JSD↓	Avg. Entropy
Qwen	Base	0.243	0.276	0.604
	Direct	0.208	0.206	0.943
	Binned	0.095	0.210	0.940
Llama	Base	0.187	0.218	0.869
	Direct	0.201	0.205	0.944
	Binned	0.278	0.210	0.963

Table 2: Results of fine-tuning experiments. **Green**: improvement from base, **Red**: deterioration from base. Fine-tuning was able to minimize the target JSD in all settings, Dist.-Corr. improved using Llama and deteriorated with Qwen.

Performance Fine-tuning successfully reduced the average JSD across all models and functions, validating the effectiveness of the algorithm. Even though the predicted distributions collapsed, a minor correlation persists post-fine-tuning, with Dist.-Corr. values ranging from 0.095 to 0.278—likely a result of balanced instances appearing in the test set. Interestingly, while Dist.-Corr. decreased for Qwen, it unexpectedly improved over the baseline for Llama, a shift potentially explained by the exceptionally low baseline correlation Llama exhibited initially.

5 Discussion

To identify reasons for the previous results, we analyze the possible rewards the function can output, revealing risk-averse behavior due to the non-

³<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.entropy.html>

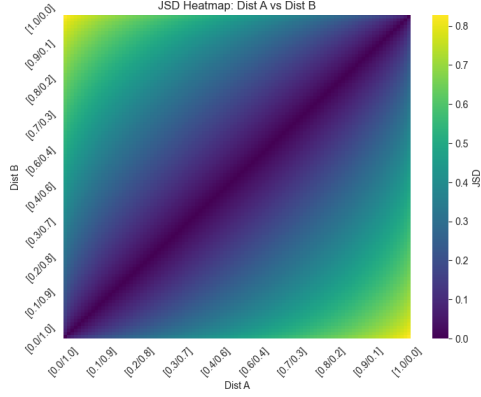


Figure 1: Values JSD can take with two-element distributions using base e . A large part of the surface is covered by values below 0.4.

uniform reward landscape of the reward functions and discuss other factors that might have contributed to the behavior.

5.1 Reward Landscape

To identify potential reasons for the failure mode, we analyzed the reward landscape, i.e., the values the rewards can take, using JSD as an optimization goal. Figure 1 shows the possible values $JSD(A||B)$ can take in a two-event case, where $A, B \in \mathcal{P}(R^2)$ are two-dimensional distributions, which is directly applicable to our binary preference task. The heatmap reveals non-uniform sensitivity: predictions near $[0.5/0.5]$ achieve $JSD < 0.4$, while extreme predictions face JSD ranging from 0.0 to 0.83 depending on the target distribution. In our case, if a model predicts highly-divergent distributions, it can either achieve very low JSD or very high JSD based on the human distribution, but if it predicts balanced predictions, no matter the target distribution, it gets a relatively good reward. This creates a risk asymmetry where conservative predictions are low-risk (bounded penalty) and extreme predictions are high-risk (variable penalty). Given this structure, RL optimization converges to near-uniform predictions that minimize expected penalty rather than accurately matching diverse human distributions.

5.2 Reasons for Failure

As previously discussed, the main issue of the method is the non-uniform sensitivity of the metric, as predictions close to the corners of the graph run the risk of producing larger errors, than those closer to the corners of the graph in Figure 1. This leads to strategies focusing on the exploitation of

the balanced and safe distributions without incentivizing exploration of more divergent predictions with a chance of receiving a smaller reward. A further issue might be RLOO’s leave-one-out baseline subtraction. When models generate diverse predictions, reward variance creates large advantages that destabilize training. Conservative predictions near $[0.5/0.5]$ yield consistent rewards and stable gradients, implicitly penalizing the exploration needed for accurate distribution matching.

5.3 Implications

Having identified what properties lead to this failure mode, we can identify other metrics and algorithms that share these characteristics and, as such, can also not be leveraged for reward design. In general, any bounded metric that creates a “safe zone” in the reward landscape, where moderate rewards can be earned across diverse targets without severe penalties, is prone to similar exploitation. Here, we focus on symmetric metrics. Appendix A.6 shows that KL-divergence, as an inherently asymmetric metric, exhibits comparable behavior. This risk-reward asymmetry makes conservative predictions a rational strategy from the model’s optimization perspective, even when they fail to achieve the intended alignment objective.

The observed behavior exemplifies reward hacking, i.e., models exploiting oversights in reward design to achieve high rewards through unintended solutions (Skalse et al., 2022). While reward hacking is well-documented in single-objective RL (Wang et al., 2023), we reveal a distinct failure mode: bounded symmetric divergences like JSD are inherently exploitable through risk aversion. Unlike typical reward hacking that exploits learned errors or specification ambiguities, our experiments suggest that JSD-based rewards fail due to their mathematical structure—the reward landscape itself incentivizes conservatism regardless of target distributions.

Our findings highlight the need to analyze reward landscapes before deployment, especially for distributional objectives where metric properties can create stable but misaligned incentives. While we showed a failure case of distribution similarity-based reward estimation, future work might try to increase the robustness of the methods by leveraging multi-objective rewards (Sorensen et al., 2024) or adding constraints to the reward.

Limitations

Our mathematical analysis focuses on two-event distributions, which directly apply to our binary preference task but may not fully characterize the reward landscape for multi-class problems. Future work should examine whether similar reward hacking emerges in higher-dimensional label spaces. Our experiments were done on a limited number of models with minimal hyperparameter-tuning, as we observed the failure after early experiments on smaller models and did not extend them to larger models. Furthermore, we discuss the usage of two distinct reward functions, which both show the described behavior, but did not fine-tune the specific hyperparameters, e.g., the size of the bins or the degree of the rewards and penalties. However, to generalize the findings to other metrics and models, more extensive tests might be beneficial. Due to computational constraints, we did not do extensive RLOO hyperparameter tuning. RLOO has previously been shown to be sensitive to failure modes under certain hyperparameters. We acknowledge that this is a contributing factor to the observed behavior, but the focus of this paper is JSD’s non-uniform sensitivity and why it contributes to the failure mode.

A possible other factor which affected the resulting failure case is the RLOO leave-one-out baseline-subtraction effect and further experiments with other optimization metrics are necessary to establish how much of the effect is attributable to the metric itself compared to the algorithm.

For most of our settings, there is a clear bias towards response A. This phenomenon is well documented and can be used to make one model seem stronger than the other simply because it came first in the order of responses (Wang et al., 2024). As we did not correlate the specific models that generated the responses to model performance, this bias did not directly impact any of our results, apart from being less aligned with human views. The dataset includes model responses from various LLMs with no models being consistently encoded as either response A or B. As such we did not randomly swap the position of the model responses for Furthermore, we aim to show the mode collapse after fine-tuning, which is still evident under a biased prediction.

Acknowledgments

This research was in parts supported by ERC Consolidator Grant DIALECT 101043235. Marie-Catherine de Marneffe is a Research Associate of the Fonds de la Recherche Scientifique – FNRS. Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Équipements de Calcul Intensif en Fédération Wallonie Bruxelles (CÉCI) funded by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region.

References

- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. [Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand. Association for Computational Linguistics.
- Beiduo Chen, Xinpeng Wang, Siyao Peng, Robert Litschko, Anna Korhonen, and Barbara Plank. 2024. [“seeing the big through the small”: Can LLMs approximate human judgment distributions on NLI from a few explanations?](#) In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14396–14419, Miami, Florida, USA. Association for Computational Linguistics.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). Preprint, arXiv:2501.12948.
- Dominik Maria Endres and Johannes E Schindelin. 2003. A new metric for probability distributions. *IEEE Transactions on Information theory*, 49(7):1858–1860.
- Wouter Kool, Herke van Hoof, and Max Welling. 2019. Buy 4 reinforce samples, get a baseline for free! *ICLR 2019 Deep Reinforcement Learning meets Structured Prediction Workshop*.
- Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.
- Junyu Lu, Deyi Ji, Xuanyi Liu, Lanyun Zhu, Bo Xu, Liang Yang, and Hongfei Lin. 2026. Aligning llm uncertainty with human disagreement in subjectivity analysis. *arXiv preprint arXiv:2605.10415*.
- Nicole Meister, Carlos Guestrin, and Tatsunori Hashimoto. 2025. [Benchmarking distributional alignment of large language models](#). In *Proceedings of*

- the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 24–49, Albuquerque, New Mexico. Association for Computational Linguistics.
- Barbara Plank. 2022. The “problem” of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.
- Guangming Sheng, Chi Zhang, Zilinfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys ’25, page 1279–1297, New York, NY, USA. Association for Computing Machinery.
- Anand Siththaranjan, Cassidy Laidlaw, and Dylan Hadfield-Menell. 2024. Distributional preference learning: Understanding and accounting for hidden context in RLHF. In *The Twelfth International Conference on Learning Representations*.
- Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward hacking. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS ’22, Red Hook, NY, USA. Curran Associates Inc.
- Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. 2024. Position: a roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning*, ICML’24. JMLR.org.
- Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. 2007. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769 – 2794.
- Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.
- Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? exploring the state of instruction tuning on open resources. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2025a. HelpSteer2-Preference: Complementing Ratings with Preferences. *arXiv preprint*. ArXiv:2410.01257 [cs].
- Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2025b. Helpsteer2-preference: Complementing ratings with preferences. In *The Thirteenth International Conference on Learning Representations*.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.
- Rushikesh Zawat, Prabhdeep Singh Sethi, and Roshan Roy. 2024. Jensen-shannon divergence in safe multi-agent rl. In *The Second Tiny Papers Track at ICLR 2024*.
- Michael JQ Zhang, Zhilin Wang, Jena D. Hwang, Yi Dong, Olivier Delalleau, Yejin Choi, Eunsol Choi, Xiang Ren, and Valentina Pyatkin. 2025. Diverging preferences: When do annotators disagree and do models know? In *Forty-second International Conference on Machine Learning*.

A Appendix

A.1 Prompt

The prompt used for fine-tuning, where the model is tasked to first output a reasoning path and the probability distribution:

Please carefully and fairly base your selection on the comments below to estimate which model-generated output, labeled 'Response A' and 'Response B', is preferable given the provided query. Base your decision on the criteria: helpfulness, correctness, coherence, complexity, and verbosity. Please first give a reasoning detailing how you come to the decision, marked with the tag 'REASONING:'. After the reasoning steps, generate a probability distribution, for how likely each response is to be chosen, marked with the tags 'PROBABILITY A:' and 'PROBABILITY B:'

Instructions: {INSTRUCTIONS}

Response A: {RESPONSE A}

Response B: {RESPONSE B}

The output should look like this:

REASONING:

PROBABILITY A:

PROBABILITY B:

Answer:

A.2 Technical details

Fine-tuning with RL algorithms requires a large amount of compute due to multiple models being necessary in the training objective. Two A100 GPUs were used to fine-tune the model for 3 epochs. Lora was used for more efficient fine-tuning with rank 8 and alpha 32. The experiments were implemented using the verl-library⁴ (Sheng et al., 2025), which supports a plethora of Reinforcement Learning algorithms for model alignment. For fine-tuning the following hyperparameters were used: learning rate was set to 1e-6, batch size of 12, rollout of n=4 and the kl-coefficient for RLOO to 0.001.

⁴<https://github.com/volcengine/verl>

A.3 Distributions

This section presents the distributions of both the test data and the predicted distributions. After fine-tuning, there is a considerable shift towards balanced distributions.

A.4 Test Data

Distribution of the human annotations in the test data.

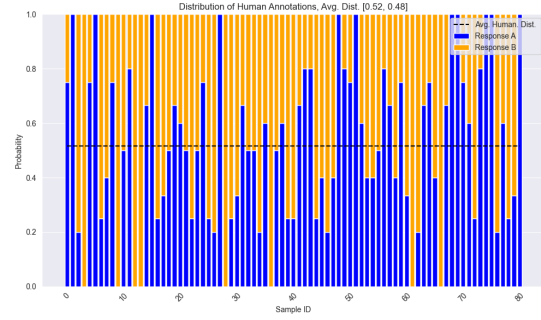


Figure 2: Distribution of the ratios of the human annotations for the valid test data. There is a wide variety of distributions in the test set.

A.5 Model Distributions

Figure 3 shows the predicted distributions on the test set of the raw and fine-tuned models. All models converge to near-uniform distributions.

A.6 Discussion of Asymmetric Metrics

In our study, we analyze the symmetric, bounded JSD and explain how it can generalize to other metrics, which show large "safe zones" in the reward landscape. Apart from such symmetric measures, other asymmetric ones warrant discussion, as their behavior might slightly differ. A common example of asymmetric divergence measures is the Kullback–Leibler divergence (Kullback and Leibler, 1951), which quantifies the information loss when one probability distribution is used to approximate another.

Running an analysis of its reward landscape (Figure 4) under the same two-event distribution setting as in our study reveals a very large "safe zone" with small values ranging from balanced distributions to more divergent ones. However, similar to JSD, there are high peaks in the corners where divergent predictions run the risk of achieving low rewards. This shows that the same issue persists in the asymmetrical metric, with the model preferring balanced distributions to avoid risking low rewards, albeit

less extreme than in JSD due to the large "safe zone". This suggests that our claim also generalizes to asymmetric distribution measures.

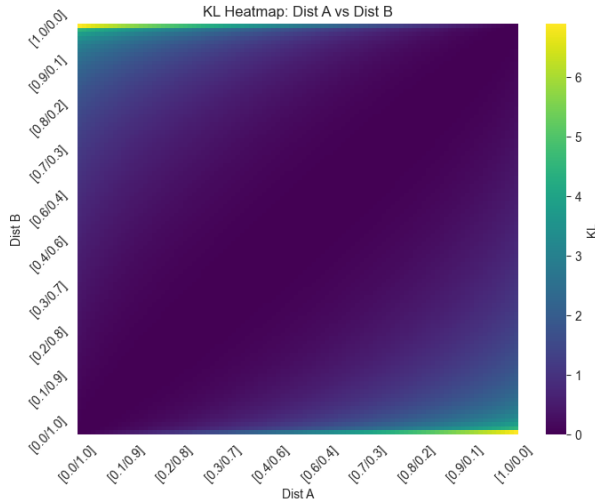


Figure 4: Values KL-divergence can take with two-element distributions. Similarly to the JSD reward landscape, the surface is covered by low values.

A.7 Logit-based probability expression

A common alternative to verbalized probabilities is the logit-based expression method, where a probability distribution is inferred by extracting specific logits from the model’s distribution over the vocabulary and normalizing it. Using the direct approach, we fine-tuned Qwen3.1-1.7B on the complete helpsteer dataset. Prior to fine-tuning, the model substantially overestimated one of the labels (avg. entropy of predictions = 0.04), and this behavior remained essentially unchanged following fine-tuning. Moreover, neither of the evaluation metrics exhibited any improvement after fine-tuning, suggesting that the adopted approach did not effectively alter the model’s behavior. A plausible explanation is that, because the initial model policy was strongly biased toward unbalanced label distributions, the model primarily received rewards for predictions corresponding to these unbalanced instances. Consequently, it failed to adequately explore more balanced distributions and instead continued to prioritise the unbalanced cases. For this experiment, the learning rate and the KL-coefficient in RLOO were set to $1e-5$ and 0.0005, respectively.

Version	Dist.-Corr.↑	Avg. JSD↓	Avg. Entropy
Base	0.04	0.43	0.04
Direct	0.04	0.44	0.08

Table 3: Results of the fine-tuning experiments. The low mean entropy shows that the predicted outcomes were highly unbalanced.

A.8 Sample

Table 4 shows a sample from the test set. The prompt is composed of instructions on how to solve the task, two model outputs from distinct models, and a template of how to format the output. The lower part details the models’ predictions after fine-tuning for both approaches and shows the JSD.

Distribution of Predicted Probabilities

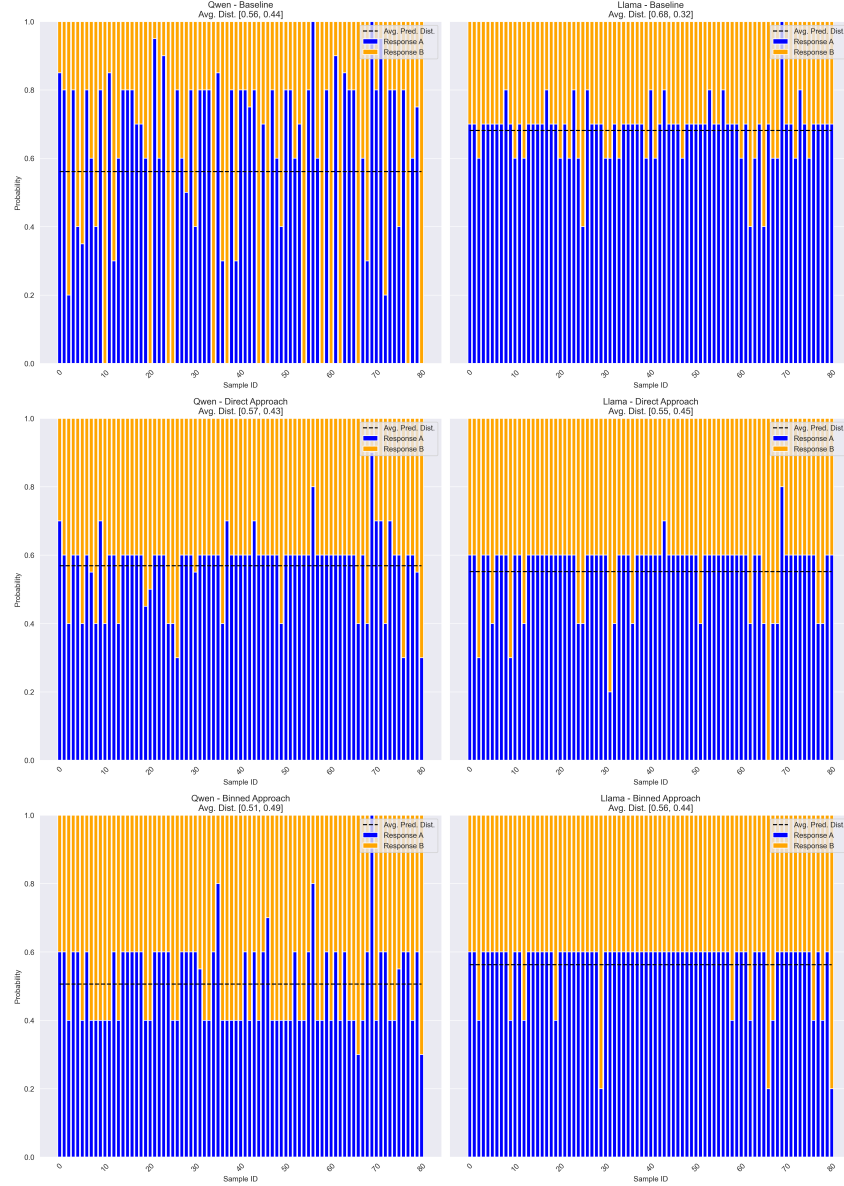


Figure 3: Overview of predicted Distributions. Each bar is made up of **Response A** at the bottom and **Response B** at the top. The average prediction is provided per model and highlighted in the plots as a dotted line. The Llama models and Qwen-direct tended to predict a [0.6/0.4] distribution, and Qwen-bins predicted [0.4/0.6].

Prompt			
Please carefully and fairly base your selection on the comments below to estimate which model generated output, labeled 'Response A' and 'Response B', is preferable given the provided query. Base your decision on the criteria: helpfulness, correctness, coherence, complexity, verbosity. Please first give a reasoning detailing how you come to the decision, marked with the tag REASONING:. After the reasoning steps, generate a probability distribution, for how likely each response is to be chosen, marked with the tags 'PROBABILITY A:' and 'PROBABILITY B:' Instructions: Make it to an academic writing setting. The observations to be made of the experiment was through data analysis of the data. that was that as depth or height increased so to as the pressure value. In the experiment involving the center of gravity apparatus, as the water, weight, and depth are proportionally direct to each other. Response A: Sure, here's a revised version of your sentence that is more appropriate for academic writing: As the depth or height of the experiment increased, so did the pressure value. In the experiment involving the center of gravity apparatus, the water, weight, and depth were proportionally direct to each other. Response B: In the experiment involving the center of gravity apparatus, the data analysis showed a direct proportionality between water weight, depth, and pressure value. As the depth or height increased, so did the pressure value. This is in accordance with the theory that water weight and depth are directly proportional to each other. This experiment provides evidence to support the idea that as the depth of a liquid increases, so does its pressure value. This is due to the fact that the weight of the liquid is distributed over a larger area, resulting in a greater pressure. This experiment also provides evidence to support the idea that the center of gravity of an object can be determined by measuring the pressure values at different points on the object. The output should look like this: REASONING: PROBABILITY A: PROBABILITY B: Answer:			
Method	Predicted Dist.	Human Dist.	JSD↓
Direct	[0.3, 0.7]	[0.2, 0.8]	0.0819
Binned	[0.4, 0.6]	[0.2, 0.8]	0.1554

Table 4: Random sample taken from the test-set of a formatted prompt containing the task and both model outputs. The predicted distribution and model distributions are presented along with the achieved JSD.

Controlled Fine-Tuning for Two-Tower Recommenders: Why Gradual Unfreezing Can Fail and Cosine Regularization Suffices

Gabriel Dershowitz

Yeshiva University

gabrieldershowitz@gmail.com

Djallel Bouneffouf

IBM Research

Djallel.Bouneffouf@ibm.com

Abstract

Controlled fine-tuning of pretrained text encoders—via gradual layer-wise unfreezing and anchoring—is widely assumed to benefit downstream tasks. We test this assumption in two-tower recommendation and, across three seeds, find a mixed picture. **First**, any fine-tuning dramatically beats a frozen encoder ($\geq 3.0\times$ cold Recall@10 lift); freezing the item tower is strongly counterproductive. **Second**, gradual unfreezing lowers warm-setting performance relative to vanilla fine-tuning under realistic budgets (12 epochs), with no measurable cold-start benefit. **Third**, our adaptive cosine-anchored regularizer shows no measurable benefit over vanilla fine-tuning, and the full adaptive recipe underperforms simpler alternatives. Thus, for two-tower retrieval, vanilla fine-tuning is a strong, low-complexity default.

1 Introduction

Two-tower recommenders separately encode users and items into a shared embedding space and have become standard for large-scale retrieval. A natural extension replaces the item tower with a pretrained text encoder to inherit semantic priors and improve generalization to rare items. A consistent empirical finding, however, is that frozen pretrained encoders in DSSM-style models are routinely beaten by simple ID-based matrix factorization on warm recommendation (Yuan et al., 2023).

We frame this as an *alignment problem* between towers: the user and item towers must converge on a shared geometry, and freezing one forces the other to contort itself to match a fixed external geometry never optimized for the recommendation signal. The obvious remedy—unfreeze and fine-tune end-to-end—induces catastrophic forgetting of the very prior that motivated using a pretrained encoder (Howard and Ruder, 2018).

We investigate whether *controlled unfreezing*—exposing the item tower’s capacity to adapt gradu-

ally, anchoring its representations to the pretrained state, and modulating the anchoring strength using validation feedback—is the right tool. Our method, Adaptive E5, combines three ingredients: (1) gradual layer-wise unfreezing of the E5 encoder triggered by NDCG plateaus—a recsys adaptation of ULMFiT (Howard and Ruder, 2018); (2) adaptive cosine-anchored regularization with a feedback-driven coefficient λ and cooldown windows around unfreeze events; and (3) a projection head identical to the frozen-encoder baseline’s, so the only difference is the fine-tuning procedure.

We evaluate on Yelp, Amazon Video Games, and MovieLens-1M under warm and inductive cold-start protocols adapted from A-LLMRec (Kim et al., 2024). Our method improves consistently over Frozen E5 on every dataset and regime, with the largest gains on the most text-rich domain. Our ablation further shows that *any* fine-tuning dramatically beats frozen encoders, and that cosine regularization alone is the strongest single component. that no single component of our adaptive recipe reliably outperforms vanilla fine-tuning—though gradual unfreezing alone yields the best cold-start numbers we observe, at a cost to warm performance.

Negative results contribution

This paper provides three findings that align with the Insights 2026 workshop theme. First, we show that the common practice of keeping the item tower frozen (Yuan et al., 2023) is strongly counterproductive: any fine-tuning yields large gains. Second, gradual layer-wise unfreezing—a technique many practitioners would reasonably try—lowers warm-setting performance under typical training budgets, with no measurable cold-start benefit. Third, our ablation finds no component of the adaptive recipe that reliably outperforms vanilla fine-tuning; the full adaptive schedule underperforms simpler alternatives. These results complicate the assump-

tion, inherited from language-model fine-tuning (Howard and Ruder, 2018), that controlled unfreezing straightforwardly transfers to retrieval-style tasks.

2 Related Work

MoRec (Yuan et al., 2023) shows that frozen-encoder DSSM models trail ID-based MF on warm recommendation, while sequential backbones benefit more reliably from textual input. A-LLMRec (Kim et al., 2024) addresses cold-start by aligning a pretrained recommender’s collaborative signal with an LLM, showing text helps most when interaction signal is sparse. We hold the two-tower architecture fixed and ask whether a careful fine-tuning recipe alone closes the gap MoRec documents.

Beyond architecture-level comparisons, the fundamental challenge of balancing adaptation with preservation of prior knowledge has been extensively studied in contextual bandit and recommender systems research (Lin et al., 2018; Bouneffouf et al., 2017; Balakrishnan et al.). These works highlight that adaptive algorithms must carefully manage the trade-off between incorporating new data and retaining previously learned representations—a tension that directly informs our investigation of controlled fine-tuning strategies.

ULMFiT (Howard and Ruder, 2018) introduced gradual unfreezing and discriminative learning rates for language model fine-tuning, motivated by catastrophic forgetting under uniform fine-tuning. Subsequent regularized fine-tuning approaches anchor parameters or representations to the pretrained state. Our method adapts both ideas to recommendation: the unfreezing trigger is validation NDCG, and the anchor coefficient is itself adaptive.

3 Method

Architecture. Let $\mathcal{U}, \mathcal{I}$ denote user and item sets. The user tower is a learned embedding table $\mathbf{u}_u = W_u \mathbf{e}_u \in \mathbb{R}^d$ with $d=128$. The item tower is $\mathbf{e}_i = g_\phi(f_\theta(t_i))$, where t_i is the item’s textual description (title + category meta-data), f_θ is `intfloat/e5-base` (12 layers, 768-dim, 110M params), and g_ϕ is a two-layer GELU MLP (768→512→128). Both outputs are L2-normalized, so $s(u, i) = \hat{\mathbf{u}}_u^\top \hat{\mathbf{e}}_i$. The head g_ϕ is architecturally identical to the one used in our Frozen-E5 baseline.

Training objective. We use symmetric in-batch InfoNCE with temperature $\tau=0.07$: $\mathcal{L}_{\text{NCE}} =$

$\frac{1}{2}(\mathcal{L}^{u \rightarrow i} + \mathcal{L}^{i \rightarrow u})$, where

$$\mathcal{L}^{u \rightarrow i} = -\frac{1}{B} \sum_{k=1}^B \log \frac{\exp(\hat{\mathbf{u}}_k^\top \hat{\mathbf{e}}_{i_k} / \tau)}{\sum_j \exp(\hat{\mathbf{u}}_k^\top \hat{\mathbf{e}}_{i_j} / \tau)}. \quad (1)$$

To anchor the item tower to its pretrained state, we keep a frozen snapshot f_{θ_0} taken before any fine-tuning and add a cosine drift penalty:

$$\mathcal{L}_{\text{reg}} = \frac{1}{B} \sum_{k=1}^B (1 - \cos(f_\theta(t_{i_k}), f_{\theta_0}(t_{i_k}))). \quad (2)$$

The total loss is $\mathcal{L} = \mathcal{L}_{\text{NCE}} + \lambda \mathcal{L}_{\text{reg}}$, where λ is time-varying.

Adaptive λ schedule. Let $\delta_e = \text{NDCG@10}(e) - \text{NDCG@10}(e-1)$. We multiply λ by 0.9 when $\delta_e > 0$ and by 1.1 otherwise, clipped to $[\lambda_{\min}, \lambda_{\max}] = [0.01, 100]$ with $\lambda_0 = 1.0$. When validation improves we relax the anchor; when it regresses we tighten it. Around each unfreeze event we hold λ fixed for $C=2$ epochs (cooldown) to avoid over-correcting during the optimization-landscape shift.

Gradual unfreezing. We unfreeze the E5 transformer one layer at a time, top-down. Initially all 12 layers are frozen; only g_ϕ is trainable. A layer patience counter c increments each epoch validation NDCG@10 fails to exceed its running best. When $c \geq \text{LAYER_PATIENCE}=5$, we unfreeze one more layer, reset c , and begin the cooldown. Unfrozen layers use $\eta_{\text{enc}}=10^{-4}$ vs. $\eta_{\text{user}}=10^{-3}$ for the user table and head, following ULMFiT. Training halts when all layers are unfrozen and $c \geq \text{LAYER_PATIENCE}$, or when a global patience counter reaches $\text{GLOBAL_PATIENCE}=3$.

Hypothesis: fine-tuning’s benefits without its risks

We hypothesize that combining gradual unfreezing with adaptive cosine-anchored regularization lets Adaptive E5 capture the benefits of fine-tuning while limiting its principal risk: catastrophic forgetting of the pretrained encoder’s prior. Gradual unfreezing preserves that prior early in training and only gradually exposes new capacity for task-specific adaptation; the cosine anchor keeps representations close to their pretrained state throughout, relaxing when validation improves and tightening when it regresses. We therefore expect Adaptive E5 to outperform both extremes: the frozen encoder, which cannot adapt at all, and vanilla end-to-end

fine-tuning, which adapts freely but risks overwriting the prior that motivated using a pretrained encoder in the first place.

4 Experimental Setup

Datasets. **Yelp Open Dataset:** top 10,000 users $\times$ 5,000 businesses; item text is name + categories + location attributes. **Amazon Video Games (Hou et al., 2024):** top 10,000 $\times$ 5,000; item text is title + category metadata. **MovieLens-1M:** full ($\approx 6,000 \times 3,700$); item text is title + genres. ML-1M has weak item text relative to Yelp and Amazon. All datasets use a minimum-interaction filter of 5 for users and items.

Split and metrics. Leave-one-out temporal split: each user’s most recent interaction is the test item, the next is validation, the rest are training. We rank against the full corpus with seen training items masked rather than sampled negatives (Krichene and Rendle, 2020), and report Recall@10 (R@10) and NDCG@10 (N@10).

Cold-start protocol. Training is unchanged across all methods. At evaluation only, we sample 15% of test users with ≥ 4 training interactions, retain $K=3$ random training items per sampled user, and re-derive the user representation from those K items. Two-tower methods use a cold user vector $= \ell_2\text{-norm}(\frac{1}{K} \sum_j \hat{\mathbf{e}}_{ij})$; SASRec consumes the K items in temporal order and uses the final hidden state. The cold pool is fixed by COLD_SEED=42 so cross-seed variance reflects training randomness only.

Baselines. **BPR-MF** via RecBole (Zhao et al., 2021); **Two-Tower (ID):** our two-tower architecture with a learned item-ID table ($d=128$); **Frozen E5** (primary control): identical architecture and loss to Adaptive E5 but with the encoder frozen; **SASRec** (Kang and McAuley, 2018); **Adaptive E5** (ours).

Training. All two-tower models use AdamW, batch 512, gradient accumulation over 2 steps, weight decay 10^{-2} , $\tau=0.07$, $\eta_{\text{user}}=10^{-3}$, $\eta_{\text{enc}}=10^{-4}$. SASRec: dim 64, 2 layers, 1 head, max sequence length 50, 30 epochs. All stochastic results are means over $n=3$ seeds.

5 Results

Warm setting. SASRec dominates warm recall everywhere, peaking at R@10 = 0.250 on

MovieLens-1M—roughly $4\times$ the next-best method. Among text-encoder methods, Adaptive E5 substantially outperforms Frozen E5: +119% warm R@10 on Yelp, +81% on MovieLens, and +5% on Amazon. On Amazon—sparse interactions, rich text—Adaptive E5 also surpasses ID-only in warm R@10 (0.027 vs. 0.021), suggesting text semantics help when collaborative signal is thin.

Cold-start. The primary controlled comparison is Adaptive E5 vs. Frozen E5: same architecture, same data, same cold pool—only the fine-tuning procedure differs. Adaptive E5 improves cold-start R@10 by $3.1\times$ on Yelp (0.030 ± 0.001 vs. 0.010 ± 0.002), +42% on Amazon (0.026 vs. 0.018), and +18% on MovieLens (0.041 vs. 0.035). In absolute terms, Adaptive E5 ranks third on cold R@10 behind SASRec and ID-only across all three datasets; both leverage training-time interaction patterns more directly than our mean-pooling cold-user representation. Adaptive E5 does achieve second-highest cold N@10 on MovieLens-1M (0.021, behind SASRec’s 0.026).

Variance. Across three seeds, Adaptive E5 cold R@10 standard deviations are small (0.001, 0.002, 0.002 for Yelp/Amazon/ML-1M), indicating stable learning dynamics. ML-1M warm performance shows higher seed variation (± 0.006), likely reflecting sensitivity to user-embedding initialization on the larger user set.

6 Ablation

Table 2 ablates each component of Adaptive E5 on Yelp—the dataset with both the largest controlled improvement over Frozen E5 (Table 1) and the richest item text among the three we evaluate—using a 12-epoch budget shared across all rows. We concentrate the ablation here because a larger overall effect and a stronger text signal give the decomposition the best chance of separating real component-level differences from seed noise; on Amazon or MovieLens, where the overall gain is smaller, the same decomposition would likely be dominated by the variance we already observe in Table 2.

Negative findings from the ablation

Any fine-tuning $\gg$ frozen. The most striking result is that any fine-tuning yields at least a $3.0\times$ cold R@10 lift over Frozen E5 (0.030–0.033 vs. 0.010 across all four fine-tuned variants). The com-

Table 1: Recall@10 (R) and NDCG@10 (N) under warm (full-corpus, seen-item masked) and inductive cold-start ($K=3$) evaluation. Stochastic entries are mean $\pm$ std over $n=3$ seeds; BPR-MF (RecBole) is deterministic. **Bold**: best per column; underline: second-best. † BPR-MF does not expose item embeddings for cold-start.

Method	Yelp				Amazon Video Games				MovieLens-1M			
	Warm		Cold		Warm		Cold		Warm		Cold	
	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10
BPR-MF (Zhao et al., 2021)†	0.064	0.033	—	—	0.048	0.030	—	—	0.063	0.032	—	—
SASRec (Kang and McAuley, 2018)	0.091$\pm$.001	0.046$\pm$.001	0.052$\pm$.003	0.026$\pm$.003	0.109$\pm$.002	0.060$\pm$.001	0.050$\pm$.002	0.026$\pm$.001	0.250$\pm$.003	0.141$\pm$.001	0.053$\pm$.003	0.026$\pm$.002
Two-Tower (ID)	0.045 $\pm$.001	0.021 $\pm$.001	0.038 $\pm$.004	0.016 $\pm$.001	0.021 $\pm$.001	0.010 $\pm$.001	0.028 $\pm$.002	0.014 $\pm$.001	0.043 $\pm$.002	0.020 $\pm$.001	0.043 $\pm$.003	0.020 $\pm$.001
Frozen E5	0.017 $\pm$.001	0.008 $\pm$.001	0.010 $\pm$.002	0.004 $\pm$.001	0.025 $\pm$.001	0.012 $\pm$.001	0.018 $\pm$.001	0.009 $\pm$.001	0.021 $\pm$.001	0.010 $\pm$.001	0.035 $\pm$.002	0.017 $\pm$.001
Adaptive E5 (ours)	0.038 $\pm$.001	0.018 $\pm$.001	0.030 $\pm$.001	0.013 $\pm$.002	0.027 $\pm$.002	0.013 $\pm$.001	0.026 $\pm$.002	0.013 $\pm$.001	0.039 $\pm$.006	0.018 $\pm$.002	0.041 $\pm$.002	<u>0.021$\pm$.001</u>

Table 2: Ablation on Yelp. Mean $\pm$ std over the same $n=3$ seeds (42, 123, 456) as Table 1. The Frozen E5 and Full-method rows are the identical Yelp entries from Table 1, reused here rather than re-derived from a separate run. **Bold**: best among fine-tuned variants (A–D) on Warm, where std bands are separated; entries within 0.001 are tied. On Cold, (A)–(D) overlap within one seed-std of each other and are not reliably distinguishable at $n=3$, so none is bolded.

Configuration	Warm		Cold	
	R@10	N@10	R@10	N@10
Frozen E5 (ref.)	0.017 $\pm$.001	0.008 $\pm$.001	0.010 $\pm$.002	0.004 $\pm$.001
(A) Vanilla fine-tune	0.047$\pm$.001	0.022$\pm$.001	0.032 $\pm$.005	0.014 $\pm$.002
(B) Gradual unfreeze	0.044 $\pm$.002	0.020 $\pm$.001	0.033 $\pm$.002	0.016 $\pm$.001
(C) Fixed $\lambda=0.1$	0.047$\pm$.000	0.022$\pm$.001	0.032 $\pm$.004	0.015 $\pm$.001
(D) Full method (ours)	0.038 $\pm$.001	0.018 $\pm$.001	0.030 $\pm$.001	0.013 $\pm$.002

mon practice of keeping the item tower frozen is strongly counterproductive.

Gradual unfreezing hurts warm, with no measured cold-start benefit. Contrary to intuition from ULMFiT, gradual unfreezing (B) reduces warm performance compared to vanilla (A) ($R@10$ 0.047 $\rightarrow$ 0.044). Cold $R@10$ shows no compensating benefit: (A), (B), and (C) are statistically indistinguishable given their seed-to-seed variance (Table 2). With a 12-epoch budget, most layers remain frozen for most of training, delaying adaptation.

No tested component reliably beats vanilla fine-tuning. Fixed cosine regularization (C) is statistically indistinguishable from vanilla (A) on both warm and cold $R@10$. Our full Adaptive E5 (D) is nominally the lowest of the four variants on cold $R@10$ (0.030), though the gap to (A)/(B)/(C) is modest given their wide cold-start variance; on warm, (D) is clearly below both (A) and (C). Combining gradual unfreezing with adaptive regularization does not recover, and may compound, each component’s individual warm-side cost.

Takeaway for practitioners. Vanilla fine-tuning is a strong, low-complexity default. At this budget, gradual unfreezing costs warm performance with

no compensating benefit we could measure, and added regularization did not reliably improve on vanilla either.

7 Conclusion

We tested controlled fine-tuning for two-tower recommendation and found a mixed picture that complicates assumptions inherited from language-model fine-tuning. First, freezing the pretrained item encoder is strongly counterproductive: any fine-tuning yields at least a $3.0\times$ cold $R@10$ lift. Second, gradual layer-wise unfreezing—a technique many practitioners would reasonably try—lowers warm-setting performance under realistic training budgets (12 epochs), with no measurable cold-start benefit. Third, no tested component of our adaptive recipe reliably outperforms vanilla fine-tuning; the full adaptive recipe underperforms simpler alternatives.

Thus, for two-tower retrieval, vanilla fine-tuning is a strong, low-complexity default. Gradual unfreezing costs warm performance without a measurable compensating benefit, and added regularization did not reliably improve on vanilla fine-tuning either. This negative insight refines the conventional wisdom from ULMFiT and suggests that for retrieval-style tasks, simpler fine-tuning suffices. Whether a longer training budget (e.g., 50+ epochs) would let gradual unfreezing fully open the encoder and close its warm-side gap remains an open question we leave to future work.

References

- Avinash Balakrishnan, Djallel Bouneffouf, Nicholas Mattei, and Francesca Rossi. Using contextual bandits with behavioral constraints for constrained online movie recommendation.
- Djallel Bouneffouf, Irina Rish, Guillermo A Cecchi, and Raphaël Féraud. 2017. Context attentive bandits: Contextual bandit with restricted context. *arXiv preprint arXiv:1705.03821*.

- Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiushi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 328–339.
- Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*.
- Sein Kim, Hongseok Kang, Seungyeon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2024. Large language models meet collaborative filtering: An efficient all-round LLM-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.
- Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1748–1757.
- Baihan Lin, Djallel Bouneffouf, Guillermo A Cecchi, and Irina Rish. 2018. Contextual bandit with adaptive feature extraction. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 937–944. IEEE.
- Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? ID- vs. modality-based recommender models revisited. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 2639–2649.
- Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*.

Do Soft Prompts Have Useful Wavelet Structure? Negative Evidence In A Controlled Setup

Soumyadeep Dey* Srikanth Tenneti Bamdev Mishra
Shobhit Niranjan Mayur Datar

Microsoft India

{soumyadeep.dey, stenneti, bamdevm, shniranjan, mayurdatar}@microsoft.com

Abstract

Soft prompts are learned embedding sequences, but nothing guarantees that their token axis behaves like a signal with useful coarse and fine structure. We test that premise with the Discrete Wavelet Transform (DWT), which rewrites a sequence into approximation (coarse, low-frequency) and detail (local, high-frequency) coefficients. Wavelets help images and audio when most squared coefficient mass concentrates in the approximation band and the detail coefficients become sparse. Trained soft prompts meet neither condition.

Our primary test decomposes a converged soft prompt into DWT bands, fine-tunes one band or both, and compares against a matched-budget baseline that continues vanilla training with no DWT. A constructed positive control first confirms that the procedure recovers a target displacement when the displacement is confined to the trainable band and rejects it when the required band is frozen. On SST-2 (67K) and AG News (120K) with Haar wavelets, and on an SST-2 check with Daubechies-2, no band-selective mode reliably beats the baseline, and the two bands are interchangeable within 0.2 pts. Direct measurement of the converged prompt finds coefficient mass roughly balanced across the bands and detail coefficients far denser than classical wavelet theory expects. Related variants behave the same way. An end-to-end wavelet prompt tuner does not reliably outperform vanilla, and a token-diversity variant makes prompt tokens more diverse without improving accuracy. On 250 held-out FLAN generation examples, existing Wavelet KD and Standard KD checkpoints are indistinguishable under token F1 and ROUGE-L.

1 Introduction

Soft prompts are ordered continuous embedding sequences, but ordering alone does not imply the local smoothness or multiscale regularity that makes wavelets effective on physical signals. Their token indices do receive distinct model positions and

may develop cross-position correlations during optimization. Conversely, prompt vectors are learned parameters rather than samples of an underlying spatial or temporal process, so wavelet alignment is a plausible transfer hypothesis and not an established property. Frequency-domain transforms separate coarse signal components from local detail components. The Discrete Wavelet Transform (DWT), the Discrete Cosine Transform, and the Fast Fourier Transform are foundational signal-processing tools, applied recently to deep learning in vision (Zeng et al., 2024; Yao et al., 2022), audio (Xie et al., 2026), and gradient compression (Wen et al., 2025). We ask whether the same idea helps on learned prompt embeddings, in particular the soft prompts of Lester et al. (2021): a learnable matrix $P \in \mathbb{R}^{L \times d}$ prepended to a frozen large language model (LLM). Splitting P along the sequence dimension into approximation coefficients (hypothesized “macro-semantics”) and detail coefficients (hypothesized “micro-syntax”) is a plausible structured prior.

We test this intuition directly. We take a converged vanilla soft prompt P , apply the DWT to obtain approximation A and detail D , and then fine-tune only A , only D , or both. The reference point is continued training of P with no DWT, which we call CTRL. If the decomposition exposed useful structure, a band-selective mode should beat CTRL. The question is narrow: does the wavelet basis line up with the directions that prompt training actually uses? We do not aim to rule out wavelets in natural language processing (NLP), but to pin down a setting where the usual signal-processing intuition breaks. To this end, our contributions are the following.

1. **A direct test of DWT-band structure in trained soft prompts.** A constructed-band positive control first verifies that the procedure detects known wavelet alignment (§4.1). On real tasks, band-selective fine-tuning shows no detail-vs-approx asymmetry with Haar or with Daubechies-2 (§4.2), and direct measurement

* Corresponding author.

of the converged prompt agrees (§5).

2. **A controlled band-selective fine-tuning design** with a matched-budget continued-training baseline (CTRL), following the matched-baseline practice of Tamkin et al. (2020) and Lee-Thorp et al. (2022). No band-selective DWT mode is reliably better than CTRL on AG News, and the nominal SST-2 gains are too small to treat as reliable without replicated full-data seeds.
3. **A negative result that persists across variants**, covering an end-to-end wavelet prompt tuner, a token-diversity test, and wavelet-decomposed distillation (Appendices D–F), including an output-level check on held-out FLAN generation examples (§4.5). One secondary finding is that DWT is associated with better stability, but not higher accuracy, in a single conv-only ablation (§4.4). We close with five hypotheses about the failure mode (§5).

2 Background

Soft prompt tuning. A learnable matrix $P \in \mathbb{R}^{L \times d}$ is prepended to the input, and only P is trained (Lester et al., 2021; Li and Liang, 2021). With $L=20$ and $d=896$ (Qwen2.5-0.5B), P has 17,920 parameters.

Discrete Wavelet Transform. A 1-level Haar DWT along the sequence dimension produces approximation $A \in \mathbb{R}^{\lceil L/2 \rceil \times d}$ (low-frequency averages) and detail $D \in \mathbb{R}^{\lceil L/2 \rceil \times d}$ (high-frequency differences). The inverse transform (IDWT) reconstructs P exactly. Classical wavelet theory expects most squared coefficient mass to concentrate in A , often written as $\|A\|^2 / \|D\|^2 \gg 1$ for natural images and audio. We test whether this holds for trained soft prompts.

Frequency transforms in deep learning. Frequency transforms have shown gains in vision (Zeng et al., 2024; Yao et al., 2022), audio (Xie et al., 2026), and LLM gradient compression (Wen et al., 2025). In NLP, FNet (Lee-Thorp et al., 2022) replaces self-attention with a fixed Fourier mixer on activations, and Tamkin et al. (2020) apply spectral filtering to language-model hidden states. Both modify the forward pass rather than tune parameters through a transform, and both re-run matched non-frequency baselines, a practice we follow (§4.2).

3 Method: DWT Surgery

Given a converged soft prompt P trained on a target task, our DWT surgery procedure is:

1. Decompose: $(A, D) = \text{DWT}(P)$ using a chosen wavelet (Haar or Daubechies-2 db2).
2. Selectively fine-tune one of: detail only (freeze A , update D), approx only (freeze D , update A), or both (update A, D jointly).
3. Reconstruct: $P' = \text{IDWT}(A', D')$, fed to the frozen LLM.

All optimization details are reported in Appendix C.

Prompt order and positional encoding. We apply the DWT to the learned prompt matrix before the frozen LLM forward pass, and the IDWT returns exactly one embedding per original prompt index. Qwen2.5 then applies its ordinary position IDs and rotary positional mechanism to the resulting sequence, and we leave both untouched. The structural assumption under test is therefore that neighboring indices of the learned prompt matrix carry useful smooth or local relationships, not that the model’s positional encoding itself has wavelet structure.

Baseline (CTRL). We compare this procedure against a matched-budget baseline, CTRL, which continues training the original P for the same number of epochs with the same fresh optimizer and learning rate but no DWT or IDWT. CTRL separates the contribution of frequency-band structure from that of the learning-rate restart itself, a known late-stage refinement technique (Mosbach et al., 2021). Without it, a gain after surgery could simply be a gain from continued training.

Models and data. We use a frozen Qwen2.5-0.5B (494M parameters, $d=896$) with prompt length $L=20$, on SST-2 (67K train, sentiment) and AG News (120K train, topic classification). Vanilla prompt tuning supplies the converged starting prompt. Optimizer, precision, batch size, stopping, and boundary-handling details are consolidated in Appendix C.

4 Results

4.1 Positive Control: Detecting Constructed Band Alignment

Before interpreting a null result on task accuracy, we verify that the surgery procedure detects wavelet

alignment when it is known to exist. Starting from the converged SST-2 prompt, we build two teacher prompts by adding a displacement of norm $0.25\|P\|$ entirely to either the approximation or the detail coefficients. From the original prompt, we then optimize the CTRL, approximation-only, detail-only, and both-band parameterizations to recover each teacher.

Table 1: Constructed-band positive control, starting from the converged SST-2 prompt with a level-1 Haar decomposition. Entries are the relative reconstruction error $\|P' - P_\star\|_F / \|P_\star\|_F$ against the teacher prompt $P_\star$, where lower is better and 0 is exact recovery. Each mode optimizes for 5 epochs at 64 update steps per epoch. Values are means over 3 random perturbation draws, with standard deviations below 4×10^{-4} . CTRL and both-band modes span the full prompt space and are reference rows.

Teacher displacement	CTRL	Approx only	Detail only	Both
Approximation band	$< 10^{-6}$	$< 10^{-6}$	0.243	$< 10^{-6}$
Detail band	$< 10^{-6}$	0.242	$< 10^{-6}$	$< 10^{-6}$

The expected direction holds on all six target/draw combinations, so the implementation and the selective-freezing design can expose band asymmetry when the target is explicitly wavelet-aligned. This does not establish that downstream task behavior is naturally aligned. A separate frozen-LLM logit-matching diagnostic favored the correct band on average, but with only three draws the paired differences remained uncertain (3/3 draws for the approximation target and 2/3 for the detail target), so we use the prompt-recovery result only as an implementation-level positive control.

4.2 DWT Surgery Does Not Separate Useful Bands

Table 2 reports the surgery experiment with a 1-level decomposition, using Haar on both datasets and Daubechies-2 on SST-2.

CTRL matches every surgery mode. On AG News, which starts from a well-converged prompt, CTRL gains as much as or more than every DWT-based mode. The SST-2 Haar modes are nominally ahead of CTRL by 0.23 pts, but we do not treat a gap that small as reliable without replicated full-data seeds, and with db2 the ordering flips, so that CTRL matches the best surgery mode and beats two of the three. Taken alone, the SST-2 rows would read as a wavelet gain, whereas against CTRL they amount to at most a small and unreplicated advan-

Table 2: DWT surgery after convergence. Frozen Qwen2.5-0.5B, $L=20$, level-1 decomposition, on SST-2 (67K) and AG News (120K). Every mode fine-tunes for 5 epochs with a fresh AdamW optimizer at $\text{lr } 10^{-4}$, starting from the converged vanilla prompt of that dataset. “Vanilla cont. [CTRL]” is the matched-budget control with no DWT. Entries are accuracy in percent, and Δ is measured against the “Vanilla (conv.)” row for that dataset, which the Haar and db2 blocks share. Detail-only and approx-only update 8,960 parameters, CTRL and both update 17,920. Single run per cell, so the second decimal is not meaningful and no winner is marked.

Setting	Method	Acc (%)	Δ
SST-2, Haar	Vanilla (conv.)	89.79	—
	Vanilla cont. [CTRL]	92.43	+2.64
	Detail only (D)	92.66	+2.87
	Approx only (A)	92.66	+2.87
	Both ($A+D$)	92.89	+3.10
SST-2, db2	Vanilla cont. [CTRL]	92.66	+2.87
	Detail only (D)	92.43	+2.64
	Approx only (A)	92.55	+2.75
	Both ($A+D$)	92.66	+2.87
AG News, Haar	Vanilla (conv.)	93.08	—
	Vanilla cont. [CTRL]	93.54	+0.46
	Detail only (D)	93.50	+0.39
	Approx only (A)	93.49	+0.38
	Both ($A+D$)	93.50	+0.39

tage. No band-selective DWT mode is reliably better than CTRL, and the same qualitative pattern holds for both tested wavelets.

The two bands are interchangeable. Tuning the high-frequency band matches tuning the low-frequency band in every tested setting, with a largest gap of 0.12 pts. Both modes halve the trainable parameters from 17,920 to 8,960 and still reach CTRL’s accuracy, so at this precision there is no band asymmetry. Tuning all wavelet coefficients jointly also stays close to CTRL, which suggests that training through the DWT behaves much like training directly.

4.3 Seed Sensitivity of Wavelet-Augmented Architectures

The surgery experiment isolates the effect of the DWT on a converged prompt. We also test WD-CPO (Wavelet-Decomposed Convolutional Prompt Optimizer, Appendix D), an end-to-end prompt tuner that decomposes P by DWT, refines each band with dilated convolutions, and recombines the bands by IDWT through a learned gate. WD-CPO does not improve on vanilla on average, and it is severely seed-sensitive, a known hazard for language-model fine-tuning on small data (Dodge

et al., 2020; Mosbach et al., 2021; Reimers and Gurevych, 2017).

Table 3: Seed sensitivity. Vanilla soft-prompt tuning and WD-CPO under identical configurations at seeds 42, 123, and 456, with frozen Qwen2.5-0.5B on SST-2 (2K) and BoolQ (9.4K). Entries are accuracy in percent, and “Range” is the largest minus the smallest of the three seeds. Bold marks the better Mean $\pm$ Std per task. Per-seed winners are not marked because they flip across seeds and are not statistically separable.

Seed	SST-2		BoolQ	
	Vanilla	WD-CPO	Vanilla	WD-CPO
42	89.6	92.2	71.8	63.4
123	91.2	90.4	74.6	66.0
456	91.8	72.0	69.6	66.2
Mean $\pm$ Std	90.9$\pm$1.1	84.9 $\pm$ 11.0	72.0$\pm$2.5	65.2 $\pm$ 1.6
Range	2.2	20.2	5.0	2.8

On SST-2, WD-CPO’s spread across seeds is nearly an order of magnitude wider than vanilla’s, and it covers both a run that beats vanilla and a run far below it. A study that happened to report the favorable seed would reach the opposite conclusion. This is why we report the matched control and the full seed table.

4.4 Conv-only Collapse: DWT as Structural Regularizer

A single-run secondary finding comes out of WD-CPO. A conv-only ablation keeps the dilated-convolution refinement but removes the DWT/IDWT round-trip. On SST-2 (2K) it reaches 91.4%, then collapses to 54.4% within four epochs and never recovers, whereas WD-CPO from the same run only drifts from 91.2% to 86.0% (Fig. 1). This suggests a possible stabilizing role for the round-trip, but we do not claim a replicated effect or an accuracy gain over vanilla.

4.5 A Generative Check on Existing FLAN Checkpoints

The surgery experiments are classification tasks. To see whether the negative pattern extends to generation, we reevaluate the existing 200K-example FLAN distillation checkpoints of Appendix F by greedy generation on 250 held-out examples from the FLAN generation family. Table 4 reports output-level token F1 and ROUGE-L in place of the loss proxy used during training.

Wavelet KD is nominally ahead of Standard KD on both metrics, but both paired bootstrap intervals cross zero. FitNets is nominally lower by 0.9

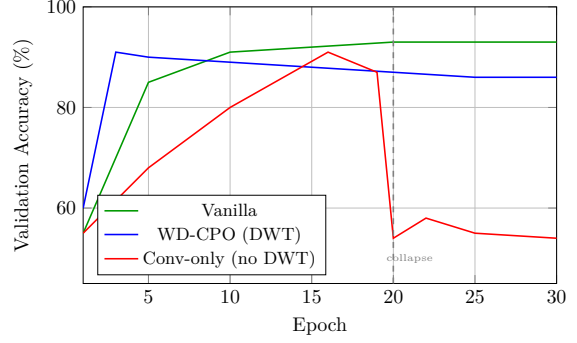


Figure 1: Training trajectories on SST-2 (2K), single run per method. Conv-only (red) peaks near 91% and collapses to 54% around epoch 20. WD-CPO with DWT (blue) drifts down instead. Vanilla (green) is stable.

Table 4: Generation metrics on 250 held-out FLAN generation examples, for the Qwen2.5-7B to Qwen2.5-0.5B distillation checkpoints of Table 9. One saved checkpoint per method, greedy decoding, 64-token limit. Entries are percentages, with metric definitions in Appendix C. The last two rows give the paired difference and its 95% bootstrap interval over 10,000 resamples of the shared examples.

Method	Token F1	ROUGE-L
Standard KD	36.25	28.59
Wavelet KD	36.86	28.81
FitNets	35.39	27.24
Wavelet – Standard	+0.61	+0.23
95% CI	[−1.05, 2.21]	[−1.34, 1.76]

and 1.4 pts. Output-level metrics therefore reach the same conclusion as the loss-based diagnostic, namely that the three distillation methods are not distinguishable here. This is evidence outside the soft-prompt setting and does not replace generative prompt surgery.

5 Why Frequency-Domain Methods Transfer Poorly Here

We organize the discussion around five hypotheses for why the pattern holds.

H1: LLM embeddings lack classical frequency structure. Wavelet theory expects energy in the approximation band and many near-zero detail coefficients. Converged vanilla prompts show neither. Writing $A/(A+D)$ for $\|A\|^2/(\|A\|^2 + \|D\|^2)$, SST-2 sits at 0.49 and AG News at 0.65, and only about 5% of detail coefficients are numerically near zero (Appendix G). That is nowhere near the sparse-detail regime that makes wavelets attractive on natural signals. Appendix J is not in tension with

this, because numerical density in vanilla prompts and post-hoc functional redundancy in WD-CPO prompts are different notions.

H2: DWT bands do not define privileged subspaces. The detail-only = approx-only equality of §4.2 is the most direct evidence, since neither band carries task-relevant signal that the other lacks. The useful degrees of freedom are not aligned with the DWT bands, and freezing either half behaves like vanilla continuation with smaller steps.

A related token-diversity check. We also test WavePT (wavelet prompt tuning, Appendix E), a T5-Base SuperGLUE variant that learns a vanilla prompt, freezes the Haar approximation coefficients, and trains only the detail coefficients. The intended mechanism works and the task result does not follow. For 100-token prompts, average pairwise token cosine similarity drops from 0.098 to 0.004, while average accuracy falls from 76.6 to 75.7 and RTE loses 2.6 pts. On CB, the one task with measurable starting redundancy, WavePT removes that redundancy entirely and accuracy does not move, and an explicit cosine-diversity regularizer fails the same way. WavePT also collapses effective prompt rank, and a rank-preserving alternating variant does not recover the accuracy either. Token redundancy is therefore not the bottleneck, and the geometry change that the wavelet step produces is not itself useful.

H3: Gradient path length may dilute the signal. Vanilla’s gradient takes one linear hop from the loss to P . The WD-CPO gradient instead passes back through the blending gate, the IDWT, four convolution layers, and the DWT. That path composes at least seven non-trivial Jacobians before it reaches the prompt. The accompanying gap is mild on easy in-distribution tasks (−1.4 pts on SST-2 full, −1.3 pts on AG News full) and large on out-of-distribution reasoning (−18.4 pts on RTE, Appendix D). We read this as a plausible optimization cost rather than a mechanism we measured directly.

H4: The auxiliary wavelet loss is small next to the main loss. In wavelet-decomposed knowledge distillation, the recorded weighted wavelet-loss components are small relative to the KD loss (Appendix F). These are loss components rather than gradient magnitudes, so we read them only as evidence that the configured auxiliary objective contributes little to the scalar training objective.

Raising β enough to change that destabilizes the main Kullback–Leibler (KL) training.

H5: Short sequences offer no multi-scale structure. $L=20$ yields sub-bands of length 10, shorter than the convolutional receptive field of 31, so multi-scale convolution is structurally degenerate. Raising the prompt to $L=100$ lifts the approximation-band mass from 0.49 to 0.80, yet accuracy still does not move and the learned WD-CPO gate weight drops (Appendix H). In these runs, longer prompts do not make the wavelet channel more useful.

What a positive result would need. A convincing positive result would have to hold against CTRL, survive multiple seeds (Table 3), and break the detail-only = approx-only symmetry by a margin replicated on the target setting. Our band-structure variants fail this test, and WavePT adds the caution that wavelet-induced diversity can appear without an accuracy gain.

6 Conclusion

The constructed positive control shows that the surgery implementation detects band asymmetry when alignment is imposed by construction (§4.1). Against that check, the CTRL comparison turns up no reliable advantage from DWT-band structure in trained soft prompts. The bands are interchangeable within 0.2 pts, and surgery gives no replicated gain over CTRL. Direct measurement agrees, because the energy is not concentrated in the approximation band and the detail band is not sparse. WD-CPO, WavePT, and wavelet-decomposed knowledge distillation are related failures rather than exceptions, and output-level FLAN generation metrics do not separate Wavelet KD from Standard KD (§4.5). Within the tested model families, prompt lengths, tasks, and level-1 wavelet configurations, the plain explanation is that the wavelet prior does not match the structure of learned prompt embeddings. The DWT round-trip may stabilize the conv-only ablation in one run (§4.4), but it buys no accuracy, and Appendix K explains why a short vanilla prompt is the relevant practical baseline for storage and key-value cache comparisons. We recommend that future spectral prompt-tuning results report a matched continued-training control.

References

- Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.
- James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón. 2022. **FNet: Mixing tokens with Fourier transforms**. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4296–4313. Association for Computational Linguistics.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. **The power of scale for parameter-efficient prompt tuning**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3045–3059. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. **Prefix-tuning: Optimizing continuous prompts for generation**. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597. Association for Computational Linguistics.
- Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations (ICLR)*.
- Nils Reimers and Iryna Gurevych. 2017. **Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging**. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 338–348. Association for Computational Linguistics.
- Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for thin deep nets. In *3rd International Conference on Learning Representations (ICLR)*.
- Alex Tamkin, Dan Jurafsky, and Noah Goodman. 2020. Language through a prism: A spectral approach for multiscale language representations. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*.
- Ziqing Wen, Ping Luo, Jiahuan Wang, Kun Yuan, Dongsheng Li, and Tao Sun. 2025. GWT: Scalable optimizer state compression for large language model training. *arXiv preprint arXiv:2501.07237*.
- Yuankun Xie, Ruibo Fu, Zhiyong Wang, Xiaopeng Wang, Songjun Cao, Long Ma, Haonan Cheng, and Long Ye. 2026. Detect all-type deepfake audio: Wavelet prompt tuning for enhanced auditory perception. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
- Ting Yao, Yingwei Pan, Yehao Li, Chong-Wah Ngo, and Tao Mei. 2022. **Wave-ViT: Unifying wavelet and transformers for visual representation learning**. In *Computer Vision – ECCV 2022*, volume 13685 of *Lecture Notes in Computer Science*, pages 328–345. Springer.
- Runjia Zeng, Cheng Han, Qifan Wang, Chunshu Wu, Tong Geng, Lifu Huang, Ying Nian Wu, and Dongfang Liu. 2024. Visual Fourier prompt tuning. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*.

Appendix Contents

A	Limitations	6
B	Ethics Statement	7
C	Experimental Setup	7
D	WD-CPO: End-to-End Wavelet-Augmented Prompt Tuning	8
E	WavePT: Wavelet Prompt Tuning for Token Diversity	8
F	Wavelet-Decomposed Knowledge Distillation	9
G	Energy Distribution in Wavelet Bands	10
H	Prompt Length and DWT Axis	10
I	Architecture Search Details	10
J	Compression Sweep Full Tables	10
K	Practical Implications: Detailed Analysis	11

A Limitations

Our main soft-prompt experiments use Qwen2.5, with a cross-architecture diagnostic on T5-Base (App. E) and a focused set of wavelet configurations chosen to keep the matched-budget comparison clean. Broader sweeps over model families, wavelet bases, decomposition depths, and other parameter-efficient methods are left to future work, and we do not claim a general impossibility result for wavelet methods in NLP. Although prompt tokens occupy ordered model positions, this study does not isolate how alternative positional mechanisms affect learned prompt smoothness. The output-level generative check uses one saved checkpoint per distillation method and 250 FLAN examples, so it is evidence outside soft-prompt surgery rather than a full generative prompt-tuning study. The positive control is synthetic and validates representational recovery under known band alignment. It does not show that the constructed displacement is a naturally occurring task direction.

B Ethics Statement

We use publicly available models (Qwen2.5-0.5B/7B) and datasets (SST-2, AG News, BoolQ, RTE, CB, FLAN), with no new data or human subjects. We see no direct path to misuse beyond those already present in soft-prompt tuning of general-purpose LLMs. Experiments ran on NVIDIA A100 80GB GPUs for a total of tens of GPU-hours, dominated by the FLAN multi-task distillation runs (Appendix F). As a negative-results paper, our intended societal contribution is to reduce redundant experimental effort across the community.

C Experimental Setup

Models.

- Soft-prompt experiments: Qwen2.5-0.5B (494M params, $d=896$, 24 layers), frozen, fp16 precision.
- Knowledge-distillation teacher: Qwen2.5-7B (7.6B params, $d=3584$, 28 layers), frozen, fp16 precision.
- Knowledge-distillation student: Qwen2.5-0.5B, trainable, fp32 precision.

Table 5: Datasets. “Train” is the number of training examples, “Cls.” the number of output classes, and “Gen.” marks free-form generation with no fixed label set. SST-2 appears both as a 2K subsample and as the full 67K split. Use codes: **S**=DWT surgery (§4.2), **W**=WD-CPO (App. D), **Z**=seed sensitivity (§4.3), **K**=knowledge distillation (App. F), which additionally uses a 10K SST-2 subsample for the axis comparison.

Dataset	Task	Train	Cls.	Use
SST-2	Sentiment	2K / 67K	2	S, W, K
AG News	Topic	120K	4	S, W
BoolQ	Yes/No QA	9.4K	2	W, Z
RTE	Entailment	2.5K	2	W
CB	NLI	250	3	W
FLAN	Multi-task	200K	Gen.	K

Datasets.

Soft-prompt training (vanilla and surgery starting point). AdamW, learning rate (lr) 10^{-3} , cosine annealing to 10^{-5} , batch 64, prompt length $L=20$, dropout 0.1, early stopping patience 10, gradient clipping $\|g\|_2 \leq 1.0$. Vanilla soft prompts have $L \cdot d = 20 \cdot 896 = 17,920$ trainable parameters.

Surgery fine-tuning. After convergence of the vanilla prompt, we run 5 additional epochs with a fresh AdamW optimizer at lr 10^{-4} . CTRL uses

the same hyperparameters with no DWT applied. Detail-only and approx-only each have $L \cdot d/2 = 8,960$ trainable parameters, while the both mode and CTRL each have 17,920.

DWT implementation. All DWT/IDWT operations use level-1 decomposition with symmetric (reflection) boundary extension along the token axis. All surgery results in §4.2 use the corrected implementation. An early db2 architecture-search row (Table 13 row 6, App. I) predates this fix and is flagged accordingly.

End-to-end WD-CPO training (Appendix D).

Same vanilla schedule but optimizing the augmented model. WD-CPO adds dilated-convolution stacks and a learnable blending scalar α , totaling 60,929 trainable parameters.

Knowledge distillation (Appendix F). AdamW with lr 2×10^{-5} , cosine annealing, batch 8 for the SST-2 checks and batch 16 for the 200K FLAN runs, and early stopping patience 5. Teacher on GPU 0 (inference only), student on GPU 1 (training). The wavelet channel adds 6.4M projection parameters (linear maps between student and teacher hidden sizes).

FLAN generation reevaluation. We reconstruct the deterministic validation split with seed 42 and evaluate the first 250 examples of its generation family. Each saved best student checkpoint is decoded greedily with a 64-new-token limit. All three metrics lowercase and collapse whitespace first, after which token F1 uses bag-of-token overlap and ROUGE-L uses the F1 form of longest-common-subsequence overlap. Exact match is 0 for all three methods. Paired 95% bootstrap intervals come from 10,000 resamples of the shared examples.

Constructed-band positive control. We start from the converged SST-2 vanilla checkpoint and apply a level-1 Haar DWT. For each of three random draws, Gaussian coefficient noise is normalized so that the reconstructed teacher displacement is $0.25\|P\|$, and it is added exclusively to either the approximation or the detail band. All modes start from the same original prompt and optimize normalized prompt mean-squared error for five epochs, with 64 update steps per epoch to match the 512-example, batch-8 behavioral budget. We also run a frozen-LLM diagnostic on 512 training and 500 evaluation examples that matches the teacher’s temperature-scaled class distribution by KL diver-

gence. Table 1 reports the prompt-recovery result, and we treat the behavioral diagnostic as exploratory because its three-draw paired differences include zero.

D WD-CPO: End-to-End Wavelet-Augmented Prompt Tuning

WD-CPO (Wavelet-Decomposed Convolutional Prompt Optimizer) augments the soft prompt with a DWT-decomposed, conv-refined, IDWT-reconstructed channel and blends it with the raw prompt via a learnable residual gate.

Architecture.

$$\begin{aligned}(A, D) &= \text{DWT}(P), \\ A' &= \text{Conv}_A(A), \quad D' = \text{Conv}_D(D), \\ \tilde{P} &= \text{IDWT}(A', D'), \\ P_{\text{out}} &= (1 - \sigma(\alpha))P + \sigma(\alpha)\tilde{P},\end{aligned}$$

where $\sigma(\cdot)$ is the sigmoid and α is a learnable scalar (initialized so $\sigma(\alpha)=0.5$). Conv_A and Conv_D are stacks of dilated 1D convolutions with dilations $\{1, 2, 4, 8\}$ along the token axis, with dropout 0.1.

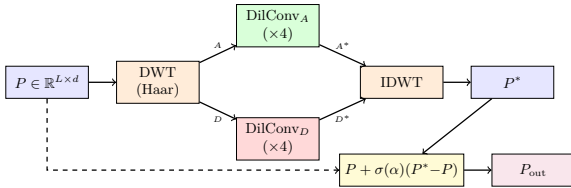


Figure 2: WD-CPO architecture. The soft prompt $P \in \mathbb{R}^{L \times d}$ is decomposed by DWT into approximation A and detail D bands, each refined by an independent dilated-convolution stack ($\text{DilConv}_{A/D}$, four layers), recombined by IDWT into $\tilde{P}$, and blended with the raw P through a learnable residual gate $\sigma(\alpha)$.

Cross-task results. The entries of Table 6 are single runs, and the seed study of §4.3 shows a ± 11 pts swing for WD-CPO on the small SST-2 split, so they should not be read as point estimates. In this set of runs, WD-CPO does not outperform vanilla anywhere. The larger gaps on the small reasoning splits (RTE, CB) match the catastrophic-basin behavior of §4.3, where a bad WD-CPO seed falls close to chance while vanilla stays stable.

α trajectory. The learned blending weight $\sigma(\alpha)$ stays at its 0.5 initialization on the small splits and rises only modestly with scale, reaching 0.58 on SST-2 67K and 0.60 on AG News. Even at 0.60,

Table 6: WD-CPO cross-task results. Frozen Qwen2.5-0.5B with $L=20$ except in the row marked $L=100$. Entries are accuracy in percent, Gap is WD-CPO minus vanilla, and bold marks the higher of the two per row. Single run per cell, not selected from multiple trials, so entries should be read against the seed sensitivity of Table 3.

Dataset	Train	Vanilla	WD-CPO	Gap
SST-2	2K	92.4	90.6	−1.8
SST-2 (full)	67K	93.4	92.0	−1.4
SST-2 ($L=100$)	67K	93.1	91.6	−1.5
AG News (full)	120K	93.1	91.8	−1.3
BoolQ	9.4K	71.7	67.3	−4.4
RTE	2.5K	72.2	53.8	−18.4
CB	250	55.4	51.8	−3.6

vanilla still beats WD-CPO by 1.3 pts, so the model learns to use the wavelet channel partially, but not enough to offset the gradient-dilution cost.

Conv-only collapse. The conv-only ablation of §4.4 runs in the same WD-CPO codebase. Fig. 1 gives the training trajectories, and rows 4 and 5 of Table 13 give the corroborating architecture-search entry.

E WavePT: Wavelet Prompt Tuning for Token Diversity

WavePT tests a related hypothesis. If standard soft prompts under-use their token dimension, then forcing local token differences through a wavelet detail band might improve both token diversity and task accuracy. This experiment is not used to establish the main DWT-surgery claim. It checks a possible prompt-geometry bottleneck.

Setup. We use frozen T5-Base (220M parameters) on BoolQ, RTE, CB, and WiC from SuperGLUE with 3 seeds (42, 123, 456), 90/10 stratified train splits, and the original validation sets as evaluation sets in a standard text-to-text prompt-tuning setup. We compare standard prompt tuning, WavePT, WavePT-Alt, and a cosine-diversity regularizer. WavePT first trains a vanilla prompt, then applies a 1D Haar DWT along the token axis, freezes approximation coefficients, and trains only detail coefficients. WavePT-Alt alternates between vanilla and detail-only stages.

Interpretation. Standard prompt tuning stays best on average. WavePT is nonetheless informative because the intended mechanism succeeds while the task result does not. It drives pairwise token cosine similarity nearly to zero across tasks,

Table 7: WavePT on SuperGLUE with frozen T5-Base and 100 prompt tokens. Entries are accuracy in percent on the SuperGLUE validation sets, reported as Mean $\pm$ Std over seeds 42, 123, and 456. “Avg” is the unweighted mean over the four tasks and is left blank for Cosine-Reg, which was not run on WiC. Bold marks the best method per column.

Method	BoolQ	RTE	CB	WiC	Avg
Standard	79.3 $\pm$ 0.3	73.0 $\pm$ 1.7	89.3 $\pm$ 1.5	64.9 $\pm$ 2.6	76.6
WavePT	79.0 $\pm$ 0.3	70.4 $\pm$ 1.8	89.3 $\pm$ 1.5	64.1 $\pm$ 1.4	75.7
WavePT-Alt	79.1 $\pm$ 0.2	71.5 $\pm$ 2.7	89.3 $\pm$ 1.5	64.0 $\pm$ 1.6	76.0
Cosine-Reg	79.1 $\pm$ 0.3	71.1 $\pm$ 0.8	88.7 $\pm$ 0.8	—	—

Table 8: Prompt-geometry metrics for the 100-token WavePT runs. Cosine is the mean pairwise token-embedding cosine similarity. Effective rank is the entropy-based rank of the prompt-token singular-value spectrum, out of a maximum of 100. Values are averaged over BoolQ, RTE, CB, and WiC, except Cosine-Reg, which excludes WiC. Bold marks the extreme value in each column, which is a nominal design target rather than a better task outcome.

Method	Cosine	Eff. Rank
Standard	0.098	86.6
WavePT	0.004	79.3
WavePT-Alt	0.022	90.8
Cosine-Reg	0.035	89.2

and on CB it removes the clearest redundancy signal (0.277 $\rightarrow$ 0.018) without changing accuracy. Unexpectedly, it also reduces effective rank, so making tokens pairwise orthogonal can still collapse them into a narrower subspace. WavePT-Alt repairs that rank loss and still does not beat standard prompt tuning. Token similarity therefore does not explain the gap, and this wavelet reparameterization changes geometry in directions that are not task-useful.

F Wavelet-Decomposed Knowledge Distillation

We additionally test whether DWT helps when distilling Qwen2.5-7B into Qwen2.5-0.5B. Standard knowledge distillation (KD) (Hinton et al., 2014) matches the student’s output distribution to the teacher’s via Kullback-Leibler (KL) divergence, and FitNets (Romero et al., 2015) additionally match intermediate hidden states via mean-squared error (MSE). We test a wavelet variant in which the student is trained with the standard KD loss (KL between teacher and student logits) plus an auxiliary loss that matches the DWT decomposition of teacher and student hidden states along a

chosen axis:

$$\mathcal{L}_{\text{KD}}^{\text{wav}} = \mathcal{L}_{\text{KD}} + \beta_A \|A_t - A_s\|_2^2 + \beta_D \|D_t - D_s\|_2^2,$$

with $\beta_A=0.1$, $\beta_D=0.1$. Hidden states are projected to a shared dimension before decomposition.

Table 9: FLAN multi-task distillation (200K samples, 10 epochs). Entries are $\exp(-\text{loss})$ on the held-out split, where higher is better. Columns are the Classification, Extraction, Reasoning, and Generation families. Each family has its own loss scale, so columns are not comparable to each other. “Avg” is the mean over the four families.

Method	Cls.	Ext.	Reas.	Gen.	Avg
Standard KD	0.051	0.094	0.344	0.165	0.163
Wavelet KD	0.051	0.091	0.345	0.165	0.163
FitNets	0.050	0.092	0.343	0.166	0.163

Multi-task FLAN. All three methods land within 0.001 of each other in every column. At the end of training, the recorded weighted wavelet-loss components are small relative to the KD loss (approximately 0.7 vs. 35). These are scalar loss-component values rather than gradient-norm measurements, consistent with the limited-objective-contribution interpretation of H4.

Axis comparison. A natural concern is the DWT axis. We retest on SST-2 (10K, 10 epochs) with the DWT applied along the embedding dimension ($d=896$) instead of the token axis ($L=128$). The

Table 10: DWT axis comparison for KD on SST-2 (10K, 10 epochs). “Best Acc” is the best evaluation accuracy reached during training, in percent. “Wav. %” is the weighted wavelet-loss components’ share of the recorded scalar objective, not a gradient-norm measurement.

Method	DWT Axis	Best Acc	Wav. %
Standard KD	—	93.00	0%
Wavelet KD	Token ($L=128$)	93.00	$\sim 2\%$
Wavelet KD	Embed. ($d=896$)	93.00	$\sim 12\%$

embedding-axis variant produces a much larger wavelet signal than the token axis, which indicates more decomposable structure along the embedding dimension. Accuracy is unchanged, so the larger wavelet signal adds nothing to what logit matching already captures.

Table 11: Energy distribution in the 1-level Haar bands of converged vanilla soft prompts (frozen Qwen2.5-0.5B, $L=20$). $\|A\|^2$ and $\|D\|^2$ are shares of the total squared coefficient mass, and $A/(A+D)$ abbreviates $\|A\|^2/(\|A\|^2 + \|D\|^2)$. The last column is the fraction of detail coefficients whose magnitude falls below the absolute threshold 10^{-3} .

Dataset	$\ A\ ^2$	$\ D\ ^2$	$A/(A+D)$	D near-zero
SST-2	49.0%	51.0%	0.49	5.3%
AG News	64.6%	35.4%	0.65	3.8%

G Energy Distribution in Wavelet Bands

Classical wavelet analysis expects energy compaction $\|A\|^2/\|D\|^2 \gg 1$ with sparse details. Trained soft prompts show neither, so the “macro-semantics in A , micro-syntax in D ” interpretation is not supported by direct measurement.

H Prompt Length and DWT Axis

Table 12: Effect of prompt length and DWT axis. The first two rows are WD-CPO runs from Table 6 and the third is the knowledge-distillation run from Table 10, so rows are not comparable to each other. Entries are accuracy in percent, with the wavelet-variant gap in parentheses. $A/(A+D)$ is the fraction of squared coefficient mass in the approximation band of the converged prompt and is undefined for the embedding-axis row. Single run per cell.

Setting	Axis	Baseline	Wavelet variant	$A/(A+D)$
WD-CPO, $L=20$	Token	93.4	92.0 (−1.4)	0.49
WD-CPO, $L=100$	Token	93.1	91.6 (−1.5)	0.80
KD, $d=896$	Embed.	93.0	93.0 (0.0)	—

Lengthening the prompt from $L=20$ to $L=100$ raises the approximation-band mass to 0.80, closer to classical wavelet structure, but the WD-CPO gap to vanilla does not shrink and the learned blending gate $\sigma(\alpha)$ drops from 0.58 to 0.47, so the model uses the wavelet channel less with longer prompts. This is the H5 observation. Switching to the embedding axis in knowledge distillation makes the wavelet signal substantial ($\sim 12\%$ of the loss) and still leaves accuracy unchanged. These are single runs, so the entries are directional.

I Architecture Search Details

Rows 1–3 below vary an auxiliary consistency-loss weight λ_c . Later rows remove that loss and vary the architectural path instead.

Key design decisions from the architecture search:

Table 13: Architecture search on SST-2 (2K train, 500 val, 30 epochs). “Best” and “Final” are the highest and last-epoch validation accuracies of the configuration, “Vanilla” is a matched-seed control run for that row, and “Gap” is Final minus Vanilla. Each row carries its own control, so rows are not comparable to each other. Bold marks the best Best and the best Final among the searched configurations, not against the controls. Row 5 (residual bypass, Haar, 1-level) is the WD-CPO configuration used throughout the paper. Single run per row. [†]Row 6 used an early db2 boundary implementation, and corrected db2 surgery results appear in Table 2.

#	Configuration	Best	Final	Vanilla	Gap
1	WD-CPO, $\lambda_c=0.5$	66.8	53.2	92.8	−26.0
2	WD-CPO, $\lambda_c=0.1$	91.6	80.0	93.0	−13.0
3	WD-CPO, $\lambda_c=0.0$	80.0	80.0	91.6	−11.6
4	Conv-only (no DWT)	91.4	54.4	93.8	−39.4
5	Resid.+Haar+1-lvl	91.2	86.0	91.6	−5.6
6	Resid.+db2+1-lvl [†]	56.0	56.0	92.0	−36.0
7	Resid.+Haar+2-lvl	87.0	77.8	91.2	−13.4
8	Resid.+Haar+ $L=32$	87.0	85.2	92.4	−7.2

- Consistency loss (λ_c) hurt training in this search (#1–3).
- Residual bypass is essential (#5 vs. #3).
- The residual Haar 1-level configuration gave the best final value among tried WD-CPO configurations. db2’s failure (#6) was likely due to a DWT implementation issue with boundary handling (subsequently fixed). In the post-hoc surgery experiment (Table 2), db2 performs comparably to Haar.
- Among tried configurations, 1-level DWT ended higher than 2-level DWT (#5 vs. #7).

J Compression Sweep Full Tables

We use A^* and D^* for the approximation and detail coefficients of the converged WD-CPO wavelet channel. The sweep below zeroes the smallest-magnitude entries of D^* post hoc, reconstructs the prompt with IDWT, and evaluates without retraining.

Accuracy is flat up to 90% sparsity on both datasets and drops only at $\geq 95\%$, so most detail coefficients in these converged WD-CPO checkpoints are functionally removable without observed accuracy loss. AG News improves slightly over most of the sweep, which we read as evaluation noise on a checkpoint whose accuracy is not limited by D^* .

Table 14: Complete D^* sparsity sweep. The smallest-magnitude entries of D^* in the converged WD-CPO prompt are zeroed post hoc, the prompt is reconstructed by IDWT, and the model is evaluated with no retraining. Entries are accuracy in percent, and Δ is relative to the 0% row of the same checkpoint. The 95% SST-2 entry is a recorded bound rather than a point estimate.

Sparsity	SST-2	SST-2 Δ	AG News	AG News Δ
0%	90.6	—	90.6	—
10%	90.6	0.0	91.6	+1.0
20%	90.6	0.0	91.6	+1.0
30%	91.4	+0.8	91.6	+1.0
40%	91.2	+0.6	92.0	+1.4
50%	91.6	+1.0	92.0	+1.4
60%	91.4	+0.8	91.8	+1.2
70%	91.0	+0.4	91.6	+1.0
80%	90.6	0.0	91.2	+0.6
90%	90.0	-0.6	92.0	+1.4
95%	≥ 89.6	≤ -1.0	90.2	-0.4
100%	87.2	-3.4	79.8	-10.8

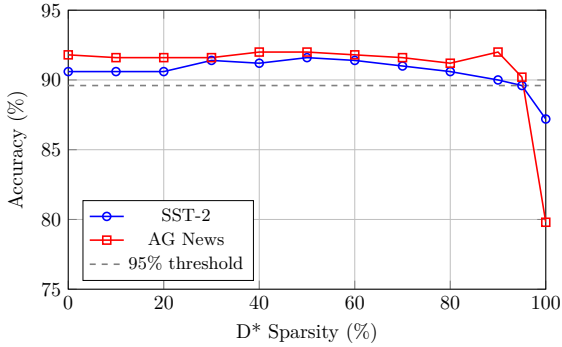


Figure 3: Accuracy versus D^* sparsity on SST-2 and AG News. Detail coefficients of the converged WD-CPO prompt are zeroed post hoc and the prompt is reconstructed by IDWT, with no retraining. Same runs as Table 14.

K Practical Implications: Detailed Analysis

This appendix supports the practical claim of §6 with a configuration-by-configuration comparison, followed by the cases where frequency-domain methods are more or less likely to transfer.

Storage and inference at matched budgets. Table 15 compares vanilla and DWT-compressed configurations on SST-2 along three practical axes. Stored floats per prompt matter for prompt-bank deployment, tokens fed to the LLM matter for key-value cache size and attention compute, and accuracy is the task outcome.

At matched storage (8,960 floats), vanilla $L=10$ beats A^* -only by +6.8 pts in this single-seed com-

Table 15: Storage, inference, and accuracy on SST-2. “Stored” is the float count needed to reconstruct the prompt, “Tokens” is what the LLM processes, “KV” is the resulting key-value cache size, and “Acc” is accuracy in percent, with bold marking the highest. WD-CPO additionally carries a 60,929-parameter convolutional module that is not amortized across tasks. The vanilla $L=20$ and WD-CPO rows come from Table 6, the DWT-compressed rows from Table 14, and vanilla $L=10$ from the Appendix C recipe with only the prompt length changed. The last row is a configuration we did not run, marked N/A. No row is a seed-averaged estimate.

Configuration	Stored	Tokens	KV	Acc
Vanilla $L=20$ (converged)	17,920	20	20	92.4
Vanilla $L=10$ (single seed)	8,960	10	10	94.0
WD-CPO $L=20$ (full D^*)	17,920	20	20	92.0
+ 95% D^* zeroed	$\sim 9,400$	20	20	≥ 89.6
+ 100% D^* zeroed (A^* only)	8,960	20	20	87.2
A^* fed directly (no IDWT)	8,960	10	10	N/A

parison. At matched token count (10 key-value cache entries), vanilla $L=10$ is the only measured configuration that achieves the saving without going through IDWT.

Why A^* cannot be fed directly to the frozen LLM. Feeding A^* as the prompt is poorly motivated without an additional adapter. The coefficients are weighted averages of adjacent embeddings, so they live in wavelet space rather than in the input distribution that the frozen first layer was trained on, and using them directly would require unfreezing layer 0 and giving up the frozen-model premise of soft prompting. The A^* -only row of Table 14 already runs IDWT and still loses 3.4 pts on SST-2 and 10.8 pts on AG News. Removing IDWT would likely widen that gap, since IDWT is the structure-preserving map from wavelet space back to embedding space, although the direct route remains untested. A vanilla $L=10$ prompt is the natural baseline in any case, because it reaches the same token count and key-value cache with no architectural overhead and higher accuracy in this single-seed comparison.

A possible follow-up is a learned wavelet token-pooling adapter, $P(20) \rightarrow \text{DWT} \rightarrow A^*(10) \rightarrow \text{Adapter} \rightarrow \text{LLM}(10 \text{ tokens})$. This reduces to learned token pooling with a fixed DWT initialization. Mean pooling and attention pooling would likely perform comparably or better, because the measurements of §5 and Appendix G indicate no advantage specific to wavelet bases over arbitrary orthogonal poolers in this regime.

When frequency-domain methods are likely to help. Our negative results are scoped to soft prompts of frozen LLMs. They do not imply wavelets are useless in deep learning. Based on the failure modes cataloged in §5, frequency-domain methods are most plausible where:

- the signal has genuine multi-scale frequency structure (audio spectrograms, time-series, images),
- sequences are long enough ($L \gg 10$) for multi-resolution analysis to separate scales meaningfully,
- the domain has a semantic grounding for “frequency” (e.g., temporal dynamics vs. frame detail in video).

The same logic predicts limited upside for short sequences, dense high-dimensional learned embeddings, and tasks where a simple baseline already saturates. Soft-prompt tuning has all three properties.

Orthogonal bases without frequency assumptions. The transferable signal from this paper is §4.4: an orthogonal DWT/IDWT round-trip appeared to prevent the catastrophic collapse of a DWT-free conv-only ablation without changing final accuracy. Nothing in this observation requires the basis to be a frequency basis. Other orthogonal bases (random orthogonal projections, PCA over the prompt parameter distribution) may provide the same structural-regularization effect without assuming a multi-scale signal that LLM embeddings do not have. We view that direction as more promising than additional wavelet variants.

Decomposing LLM-Judge–Human Agreement: How Much Is Inherited Surface Preference? *

Gaurav Kumar
Moveworks AI
gkumar@ucsd.edu

Abstract

LLM-as-judge evaluation is typically validated using judge–human agreement with standard benchmarks reporting agreement rates of roughly $\sim 80\text{--}85\%$. We show that this aggregate metric can be misleading. Most of that agreement comes from pairs where both judge and human prefer the longer, better-formatted response, rather than from agreement on response quality. On the remaining pairs, agreement drops to chance levels on MT-Bench, and below chance when surface cues mislead. We show this with two simple logistic-regression baselines on length, structure, and lexical features: one predicts the human’s pick, one predicts the judge’s. The pattern holds across closed and open judges and both benchmarks, and is strongest on the expert annotations the field treats as the quality reference. We suggest reporting the *inheritance share* alongside aggregate agreement: a simple reporting metric we defined that shows whether a judge agrees with humans for the right reasons.

1 Introduction

Using a strong LLM as an automatic judge of open-ended generation has become the default in model development and evaluation (Zheng et al., 2023; Liu et al., 2023; Chiang et al., 2024). It is validated by judge–human agreement: on MT-Bench, GPT-4-class judges agree with human annotators roughly 80% of the time, comparable to inter-human agreement (Zheng et al., 2023).

Length and formatting biases in LLM judges are well documented (Saito et al., 2023; Wu and Aji, 2023; Dubois et al., 2024). Humans show the same biases (Hosking et al., 2024; Watts et al., 2024), with LLM judges typically biased more on length (Dekoninck et al., 2025; MTalk-Bench Team, 2025). Taking these biases as established,

we ask instead: given that judges and humans share these biases, how much of the reported judge–human agreement is just the co-occurrence of shared surface preference, and how much is residual quality alignment? If much of the agreement is the former, the aggregate number measures correlated bias more than judge competence, and even a length-controlled judge (Dubois et al., 2024) does not solve the metric-level problem.

To make this quantitative, we train two trivial classifiers on the same surface features—length, structure, lexical—one targeting the human verdict, the other the judge’s. We decompose agreement into pairs both classifiers predict correctly (*surface-explained*) and the rest (*residual*). On MT-Bench, 85–88% of agreement sits in the surface-explained set, and on the residual it drops to chance.

Our main contribution is this decomposition and the resulting inheritance share $\pi_S \cdot \text{Agr}_S / \text{Agr}$ (detailed in Equation 2) proposed as a reporting metric. It quantifies how much of headline judge–human agreement is inherited from shared surface preferences rather than residual quality alignment. We find that the inherited component is largest on the expert MT-Bench annotations the field treats as the quality reference - the opposite of what annotator noise alone would predict - and smaller on crowdsourced Arena. We propose reporting the inheritance share alongside aggregate agreement: a simple diagnostic that can be computed from existing human labels.

2 Related Work

LLM-as-judge and its validation. MT-Bench and Chatbot Arena (Zheng et al., 2023) report $>80\%$ judge–human agreement, which the field has since used as the validity argument for the paradigm. Meta-evaluation work pushes back on this (Zeng et al., 2024; Bavaresco et al., 2025), and recent surveys cover the broader landscape (Gu

*Code is available at <https://github.com/gaurav5590/llm-judge-human-decomposition>.

et al., 2024; Li et al., 2024).

Judge-side biases. LLM judges exhibit position bias (Wang et al., 2024), self-preference (Panickssery et al., 2024), and broader cognitive biases and inconsistency (Koo et al., 2023; Stureborg et al., 2024). Verbosity and style biases on the judge side are extensively documented (Saito et al., 2023; Wu and Aji, 2023; Singhal et al., 2023), and Dubois et al. (2024) propose length-controlled regression for win rates.

Human-side bias and closest prior work. Humans show length and style biases too: Hosking et al. (2024) show human preference is not a reference, and MTalk-Bench Team (2025); Watts et al. (2024) report parallel human-judge length preferences. Closest to our work, POLYRATING (Dekoninck et al., 2025) fits a bias-aware rating system with shared surface-feature parameters on both human and LLM-judge data. They find that length bias is significant in both paradigms and larger for LLM judges. We take their parallel-bias finding as given. Our contribution is orthogonal: rather than estimating bias magnitudes in rating points, we decompose judge-human agreement into surface-explained and residual components (Eq. 2), propose the inheritance share as a reporting metric, and show it is largest on MT-Bench’s expert labels - a contrast not analyzed by Dekoninck et al. (2025). Hu et al. (2024) also propose a decomposition but of the pairwise win rate; we decompose the agreement metric itself.

Surface-cue meta-predictors and selective evaluation. Lehmann et al. (2025) fit a logistic-regression meta-predictor on surface cues for comparative QA, a different task; we adapt that methodology to LLM-as-judge and apply it to both the judge and the human side. Jung et al. (2025) and Radharapu et al. (2025) study confidence-based escalation and no-consensus items, settings where judge-human agreement is hardest to obtain.

3 Method

Data. We use two pairwise-preference benchmarks with human labels: **MT-Bench** human judgments (expert annotators) and a **Chatbot Arena** subset (crowdsourced), both from Zheng et al. (2023). We flatten conversations to assistant responses, deduplicate to unique (question, model pair) items, take the majority human label, and

drop ties. For Arena we subsample 1,000 pairs stratified by winner and length-difference quartile.

Not all pairs survive to analysis. First, pairs with no decisive human majority are dropped. Second, each pair is judged in both display orders; if a judge picks different winners across the two orders, the position-averaged verdict is a tie and the pair is dropped. Because judges differ in how often they flip, each judge yields a different number of usable pairs. This yields ~ 800 decisive MT-Bench pairs and ~ 700 – 900 decisive Arena pairs per judge (Appendix A).

Judges. We evaluate three judges: two closed (gpt-4o-mini-2024-07-18, gpt-4o-2024-11-20; OpenAI, 2024) and one open (Qwen2.5-7B-Instruct; Qwen Team, 2024), using the standard MT-Bench pairwise prompt (Appendix B) at temperature 0. Each pair is judged in both display orders. We take the position-averaged verdict and treat order-inconsistent pairs as ties. Verdicts parse cleanly (unparseable rate $\leq 1.6\%$ for every judge; Appendix C).

Surface features. From each response we compute twelve interpretable features grouped into three families: four *length* features (characters, tokens, words, sentences), five *structure* features (markdown headers, bullets, fenced code blocks, bold spans, paragraphs), and three *lexical* features (unique-word ratio, mean word length, question-response word overlap). Tokens use the cl100k_base tokenizer so counts are judge-independent.

For each pair (A, B) , we encode each feature as a signed difference $\Delta_f = f_A - f_B$ and, for length and structure features, additionally as a ratio $f_A/(f_A + f_B)$. The classifier sees only these Δ and ratio encodings. By construction this representation is *position-symmetric*: swapping A and B negates the feature vector and flips the prediction accordingly, so the baseline carries no positional preference of its own - in contrast to the judges.

The baseline model. Our baseline is deliberately the simplest competent classifier: ℓ_2 -regularized logistic regression on standardized features ($C=1$, no hyperparameter search; Pedregosa et al., 2011). We fit two instances with identical inputs but different targets: **Baseline-H** predicts the human verdict and **Baseline-J** predicts the judge verdict. Both use 5-fold stratified cross-validation, with out-of-fold predictions for every pair so the decomposition

Data	Judge	B-H	B-J	J-H	Flip
MT-B	4o-mini	0.804	0.868	0.854	0.141
	4o	0.803	0.865	0.852	0.119
	Qwen-7B	0.807	0.852	0.840	0.312
Arena	4o-mini	0.686	0.771	0.842	0.101
	4o	0.686	0.737	0.844	0.091
	Qwen-7B	0.716	0.745	0.830	0.259

Table 1: Surface baseline predicting humans (B-H) and judges (B-J) vs. judge–human agreement (J-H), and position-flip rate (Flip). Bootstrap 95% CIs have half-width ≈ 0.025 . B-H tracks B-J across all six cells.

uses no in-sample information.

All analyses are restricted to pairs decisive for both judge and human. We report bootstrap 95% confidence intervals over 1,000 resamples. Because the model is linear and deliberately untuned, any predictive power it shows is a lower bound on what surface cues afford.

Inheritance decomposition. We partition pairs into those the trivial baseline correctly predicts on both sides (*surface-explained*) and those it does not (*residual*), then ask what fraction of agreement lives in each. Let h_i, j_i be the human and judge verdicts on pair i and $\hat{h}_i, \hat{j}_i$ the corresponding out-of-fold baseline predictions. Define

$$S_i = \mathbb{1}[\hat{h}_i = h_i \text{ and } \hat{j}_i = j_i], \quad \pi_S = \frac{1}{N} \sum_i S_i. \quad (1)$$

Let $\text{Agr}(j, h) = \frac{1}{N} \sum_i \mathbb{1}[j_i = h_i]$ be the overall judge–human agreement rate, and let $\text{Agr}_S, \text{Agr}_R$ denote the same quantity restricted to pairs with $S_i = 1$ and $S_i = 0$, respectively. Then

$$\text{Agr}(j, h) = \pi_S \text{Agr}_S + (1 - \pi_S) \text{Agr}_R. \quad (2)$$

We call $\pi_S \cdot \text{Agr}_S / \text{Agr}$ the *inheritance share*: the fraction of total agreement that a trivial baseline already accounts for.

4 Results

4.1 The baseline predicts humans almost as well as judges

Table 1 reports the two baselines and judge–human agreement. On MT-Bench, the surface baseline predicts *human* verdicts at 80.3–80.7% across all three judges - only 4–6 pp below its accuracy at predicting the judge (B-J, 85.2–86.8%). A model that sees only length and format reproduces human preferences nearly as well as a frontier LLM

Data	Judge	π_S	Agr_S	Agr_R	Agr	Inh.%
MT-B	4o-mini	.745	.976	.498	.854	85.1
	4o	.755	.990	.425	.852	87.8
	Qwen-7B	.737	.992	.415	.840	87.0
Arena	4o-mini	.628	.991	.590	.842	74.0
	4o	.602	.985	.631	.844	70.3
	Qwen-7B	.616	.989	.576	.830	73.4

Table 2: Inheritance decomposition (Eq. 2). π_S : surface-explained fraction; $\text{Agr}_S/\text{Agr}_R$: agreement on explained/residual pairs; Inh.%: inheritance share $\pi_S \cdot \text{Agr}_S / \text{Agr}$. On MT-Bench the residual collapses to chance.

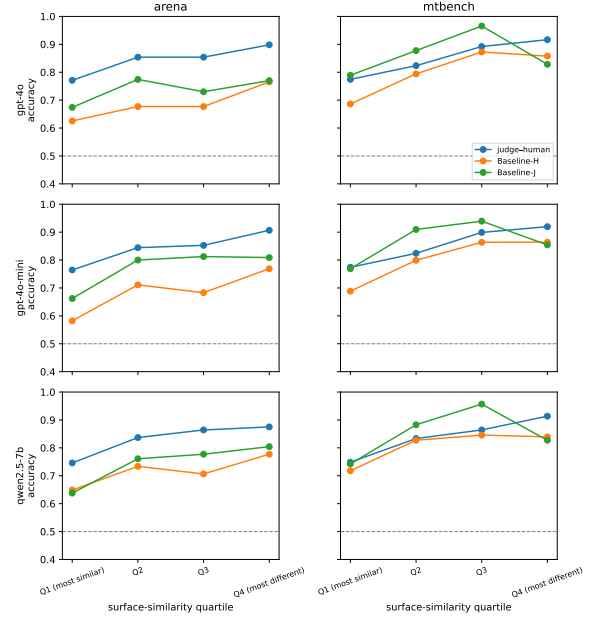


Figure 1: Accuracy by quartile of multi-feature surface distance (Q1 most similar, Q4 most different), per judge and dataset. Judge–human, Baseline-H, and Baseline-J move together; where surface cues are weak (Q1), agreement weakens too. Dashed line: chance.

agrees with those humans. We use this as input to the decomposition. On crowdsourced Arena the baseline is weaker (B-H 68.6–71.6%, still within 3–9 pp of B-J), with judge–human agreement further above B-H - a larger non-surface component than on MT-Bench.

4.2 Most agreement is the inherited surface term

Table 2 decomposes agreement via Eq. 2. On MT-Bench, $\pi_S \approx 0.74$ – 0.76 of pairs are surface-explained, agreement on them is $\approx 98\%$, and on the residual it *collapses to chance* ($\text{Agr}_R = 0.42$ – 0.50). The inheritance share thus accounts for **85–88%** of all judge–human agreement. Arena is

more tempered: fewer pairs are surface-explained ($\pi_S \approx 0.60$ – 0.63) and, crucially, residual agreement does *not* reach chance ($\text{Agr}_R = 0.58$ – 0.63). The inherited share still dominates (70–74%), but a genuine residual quality-alignment signal remains. The effect is therefore strongest exactly for the expert annotations treated as the quality reference – the opposite of what an annotator-noise account would predict. The inherited share stays high even on *objective* categories (math, coding, extraction), where extra words should not track quality (Appendix I), and is stable across filtering choices, including the aggressive position-flip drop (Appendix J). Additional robustness checks are in Appendices E, F, and G.

4.3 Agreement tracks surface separability and confidence

Figure 1 stratifies pairs by surface distance (the ℓ_2 norm of the standardized Δ -feature vector). All three curves rise together across quartiles, with judge–human agreement going from 0.75 on the most surface-similar pairs to 0.91 on the most surface-different. Agreement on the most-similar quartile does not hit 0.50; that drop only shows up on the *residual* set (Section 4.2), where the baseline is actively wrong. The point of the figure is that when surface cues are weak, human and judge agreement is weak too. Binning by baseline confidence (Figure 2) gives the same picture: agreement climbs from 0.70–0.80 in the least-confident bin to over 0.90 in the most-confident bin on both benchmarks. When the surface signal is strong, judge and human agree; when it is weak, they do not.

4.4 Humans and judges weight the same cues

The two baselines are not just accurate, they rely on similar features. Table 3 shows that token count and token ratio dominate the standardized coefficients for both targets, with the same sign and comparable magnitude. Humans and judges respond to the same length cues in the same direction, which is the mechanism the decomposition needs. Position-flip rates (Table 1) are low for the GPT judges (9–14%) but high for Qwen-7B (26–31%), consistent with smaller open judges being less order-robust (Wang et al., 2024). We position-average throughout to handle this.

4.5 Length causally moves judge verdicts

Tables 2 and 3 establish that both judges and humans rely on similar length-related cues. However,

feature	$ \bar{w}_H $	$ \bar{w}_J $
ratio_tokens	2.159	2.358
delta_chars	0.982	1.882
delta_tokens	1.178	1.752
delta_unique_word_ratio	1.010	1.409
delta_words	0.223	0.776
ratio_sentences	0.654	0.414
delta_sentences	0.641	0.354
delta_paragraph_count	0.362	0.578

Table 3: Top features by mean |standardized coefficient| for Baseline-H ($|\bar{w}_H|$) and Baseline-J ($|\bar{w}_J|$), averaged over judge $\times$ dataset. Length/verbosity dominates both, with the same sign.

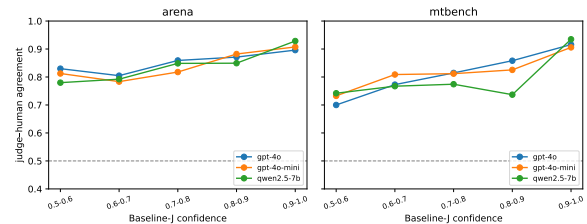


Figure 2: Judge–human agreement vs. Baseline-J confidence ($\max(p, 1-p)$), per dataset – one line per judge. Agreement increases monotonically with surface confidence on both benchmarks. Dashed line – chance.

these analyses are observational and do not show whether length itself influences judgments or simply correlates with response quality. To isolate the effect of length, we perform a verbosity-injection intervention.

We begin with the 198 MT-Bench pairs for which the judge originally preferred the *shorter* response. For each, we rewrite the longer, losing response with gpt-4o to roughly twice its length, instructing the model to add no new information and only rephrase or restate existing content. An independent model from a different family compared to all judges (claude-haiku-4-5) then verifies that no facts or claims were added. We retain the 133 rewrites that pass and grow in length, and re-judge each in both presentation orders (Appendix H).

Because content is held constant by construction, any change in verdict is attributable to length and formatting alone. Table 4 reports the fraction of pairs for which the judge now prefers the padded response, averaged over both display orders: gpt-4o-mini flips on 43%, Qwen2.5-7B on 28%, and gpt-4o on 16%, with all three confidence intervals excluding zero. Increasing verbosity alone thus reverses a substantial fraction of decisions while leaving content unchanged, and the effect is strongest for the weakest-performing

Judge	n	Flip	95% CI
gpt-4o-mini	91	.429	[.330, .538]
gpt-4o	91	.165	[.099, .242]
Qwen2.5-7B	76	.276	[.184, .382]

Table 4: Verbosity-injection results on MT-Bench. Flip is the fraction of decisive re-judged pairs for which the judge switches to the padded response. Responses are expanded to $\approx 2.2\times$ length without changing content (133 verified rewrites).

Judge	Objective		Subjective	
	Agr_R	Inh.%	Agr_R	Inh.%
gpt-4o-mini	.659	76.9	.384	89.8
gpt-4o	.578	81.7	.342	90.5
Qwen2.5-7B	.467	81.2	.407	88.9

Table 5: Inheritance decomposition pooled over *objective* (math, coding, extraction) vs. *subjective* (writing, roleplay, humanities) MT-Bench categories. The inherited share stays $\geq 77\%$ on objective tasks, where extra length cannot make a near-binary answer more correct; Per-category detail in Appendix I.

judge.

4.6 Length behaves as bias even on objective tasks

If length reflects genuine quality rather than bias, its effect should be smaller on tasks where longer responses are unlikely to be more correct. MT-Bench lets us test this by separating objective tasks (math, coding, extraction) from subjective ones (writing, roleplay, humanities), where more detailed responses may legitimately improve quality.

We recompute the decomposition for each group while keeping the global baseline fixed (Table 5). The inherited share is 8–13 pp lower on pooled objective categories, suggesting that length carries some genuine signal there. However, it remains at least 77%, indicating that most of the agreement between judges and humans is still explained by length even where verbosity is unlikely to improve correctness. Since each category contains relatively few examples ($n \approx 80\text{--}120$), we focus on the pooled comparison rather than individual categories (Appendix I).

5 Discussion

What agreement measures. A single judge–human agreement number mixes two regimes: surface-explained pairs, where judges and humans agree trivially, and the residual, where genuine

quality alignment would have to live. Reporting one aggregate hides that the headline is carried by the surface-explained regime, and that the residual where actual quality adjudication is needed can sit close to a chance probability.

An audit of the agreement metric. We propose reporting the *inheritance share* $\pi_S \cdot \text{Agr}_S / \text{Agr}$ alongside aggregate agreement, and/or stratifying by surface separability (e.g. residual agreement Agr_R). This requires only the human labels already collected and a trivial baseline; it costs nothing extra and surfaces whether a judge agrees with humans for the right reasons. It is complementary to length-controlled estimators (Dubois et al., 2024), which debias the score, and to bias-aware rating systems (Dekoninck et al., 2025), which estimate bias magnitudes on ratings; we directly audit the *agreement metric itself*. We do not claim the inheritance share is sufficient but it is a near-free signal currently missing from standard reports.

6 Conclusion

We split judge–human agreement into two parts using a simple surface baseline: pairs the baseline gets right on both sides, and the rest. On MT-Bench, 85–88% of agreement falls in the first part. Our results suggest that headline judge–human agreement largely reflects shared surface preferences rather than shared quality judgments, motivating the inheritance share as a complementary reporting metric. The inheritance share is a cheap way to see this in any evaluation with human labels.

Limitations

We test turn-1, English, pairwise preference on two benchmarks and three judges; the numbers may differ for other tasks, languages, judges, and pointwise scoring. The split into surface-explained and residual depends on the features we use; a richer set (e.g. semantic embeddings) would likely move more pairs into the surface-explained bucket, so our numbers are a lower bound on what surface cues can capture. We treat human labels as ground truth even though annotators have biases, make errors, and sometimes legitimately disagree (Plank, 2022; Hosking et al., 2024; Bavaresco et al., 2025)—our claim is that this “ground truth” shares the judge’s surface preference, not that either is correct. The Arena subset is 1,000 pairs to keep costs down, so the confidence intervals there are wider.

References

- Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, and 1 others. 2025. [LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anas-tasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. [Chatbot arena: An open platform for evaluating LLMs by human preference](#). In *International Conference on Machine Learning (ICML)*.
- Jasper Dekoninck, Maximilian Baader, and Martin Vechev. 2025. [POLYRATING: A cost-effective and bias-aware rating system for LLM evaluation](#). In *The Thirteenth International Conference on Learning Representations (ICLR)*.
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. [Length-controlled AlpacaEval: A simple way to debias automatic evaluators](#). *arXiv preprint arXiv:2404.04475*.
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. [A survey on LLM-as-a-judge](#). *arXiv preprint arXiv:2411.15594*.
- Tom Hosking, Phil Blunsom, and Max Bartolo. 2024. [Human feedback is not gold standard](#). In *The Twelfth International Conference on Learning Representations (ICLR)*.
- Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Athanasios V. Vasilakos, and Hui Xiong. 2024. [Explaining length bias in LLM-based preference evaluations](#). *arXiv preprint arXiv:2407.01085*.
- Jaehun Jung and 1 others. 2025. Trust or escalate: LLM judges with provable guarantees via selective evaluation. *arXiv preprint*.
- Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2023. [Benchmarking cognitive biases in large language models as evaluators](#). *arXiv preprint arXiv:2309.17012*.
- Jan Lehmann and 1 others. 2025. Knowing the facts but choosing the shortcut: Surface cues in comparative question answering. *arXiv preprint*.
- Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, and 1 others. 2024. [From generation to judgment: Opportunities and challenges of LLM-as-a-judge](#). *arXiv preprint arXiv:2411.16594*.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. [G-Eval: NLG evaluation using GPT-4 with better human alignment](#). In *Proceedings of EMNLP*.
- MTalk-Bench Team. 2025. MTalk-Bench: Evaluating speech-to-speech models with human and LLM judges. *arXiv preprint*.
- OpenAI. 2024. [GPT-4o system card](#). *arXiv preprint arXiv:2410.21276*.
- Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. [LLM evaluators recognize and favor their own generations](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Barbara Plank. 2022. [The “problem” of human label variation: On ground truth in data, modeling and evaluation](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Qwen Team. 2024. [Qwen2.5 technical report](#). *arXiv preprint arXiv:2412.15115*.
- Bhaktipriya Radharapu and 1 others. 2025. Arbiters of ambivalence: Challenges of using LLM judges on no-consensus tasks. *arXiv preprint*.
- Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. [Verbosity bias in preference labeling by large language models](#). *arXiv preprint arXiv:2310.10076*.
- Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. [A long way to go: Investigating length correlations in RLHF](#). *arXiv preprint arXiv:2310.03716*.
- Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. [Large language models are inconsistent and biased evaluators](#). *arXiv preprint arXiv:2405.01724*.
- Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. [Large language models are not fair evaluators](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Ishaan Watts, Varun Gumma, Aditya Yadavalli, Vivek Seshadri, Manohar Swaminathan, and Sunayana Sitaram. 2024. [PARIKSHA: A large-scale investigation of human-LLM evaluator agreement on multilingual and multi-cultural data](#). *arXiv preprint arXiv:2406.15053*.

Minghao Wu and Alham Fikri Aji. 2023. [Style over substance: Evaluation biases for large language models](#). *arXiv preprint arXiv:2307.03025*.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. [Evaluating large language models at evaluating instruction following](#). In *International Conference on Learning Representations (ICLR)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-judge with MT-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*.

A Data and Filtering

Both datasets are reduced to a normalized pair schema (question, answer_a, answer_b, human_winner). MT-Bench uses the turn-1 human-judgment split; conversations are flattened to assistant text. We drop tie and tie (bothbad) labels and pairs with a tied human vote. Arena is subsampled to 1,000 pairs stratified by winner $\times$ length-difference quartile (seed 42), English-only. Decisive-pair counts per cell range from 649 to 905.

B Judge Prompt

We use the standard MT-Bench pairwise system prompt:

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants. . . Avoid any position biases. . . Do not allow the length of the responses to influence your evaluation. . . After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

The user turn presents the question and the two answers; we parse the last bracketed token as the verdict.

C Additional Results

Verdict unparseable rates are uniformly low: gpt-4o-mini 0.0%, gpt-4o 0.4–1.6%, Qwen2.5-7B 0.0–0.1%. Per-cell decisive-pair counts (n) appear in Table 1’s caption range and in our released CSVs.

D Position Bias by Judge

Table 6 reports position-flip rates out of all human-decisive pairs (the same denominator used throughout the paper). GPT judges flip on 10–18% of pairs; Qwen flips on 31–36%. All analyses use position-averaged verdicts; inconsistent pairs are treated as ties and excluded.

E Flip-Pair Sensitivity

We drop position-inconsistent pairs as ties throughout the main analysis. Table 7 checks what changes if we count them as disagreements instead. For GPT judges the shift is small (−8 to −12 pp); for Qwen, which flips on 26–31% of pairs, inclusive agreement falls to 0.59–0.61. Either way the main

Dataset	Judge	Flip rate	Consistent
Arena	gpt-4o-mini	10.5%	89.5%
Arena	gpt-4o	13.5%	86.5%
Arena	Qwen2.5-7B	30.6%	69.4%
MT-Bench	gpt-4o-mini	14.5%	85.5%
MT-Bench	gpt-4o	18.1%	81.9%
MT-Bench	Qwen2.5-7B	36.4%	63.6%

Table 6: Position-flip rates per judge and dataset, out of all human-decisive pairs. Qwen2.5-7B is much less robust to display order than the GPT judges. These are the same Flip% values shown in Table 7.

results hold and Qwen’s agreement with humans is lower than the strict figures suggest.

Data	Judge	Flip%	Strict	Incl.
Arena	gpt-4o-mini	10.5	.842	.757
Arena	gpt-4o	13.5	.844	.764
Arena	Qwen2.5-7B	30.6	.830	.612
MT-B	gpt-4o-mini	14.5	.854	.732
MT-B	gpt-4o	18.1	.852	.750
MT-B	Qwen2.5-7B	36.4	.840	.588

Table 7: Sensitivity to treating position-inconsistent pairs as disagreements. Strict J–H is the figure used throughout the paper (flips dropped); inclusive counts them as judge–human disagreements. The gap is small for GPT judges and large for Qwen2.5-7B, confirming the strict figures are conservative.

F The Residual Collapse Is Not Mechanical

One concern is that the residual pairs are just hard and any two signals would agree at chance on them. We check this by splitting pairs into three groups: **S** (both baselines correct, same as the surface-explained set), **CW** (both baselines predict the same winner but are wrong), and **D** (baselines predict different winners).

If the collapse were mechanical, both CW and D should be near chance. Table 8 shows they are not: CW pairs on MT-Bench give J–H agreement of 0.33–0.38, *below chance*, while D pairs give 0.65–0.79. When surface cues confidently point the wrong way, judge and human actually anti-correlate. The D result (4–9% of pairs) suggests a small genuine quality signal when surface cues are ambiguous.

G Feature-Set Robustness

We rerun the decomposition with three feature sets. The ranking (MT-Bench > Arena across all judges)

Data	Judge	S	CW	D
Arena	gpt-4o-mini	.991	.530	.828
Arena	gpt-4o	.985	.551	.835
MT-B	gpt-4o-mini	.976	.357	.730
MT-B	gpt-4o	.990	.378	.654

Table 8: J-H agreement by group (Qwen follows the same pattern). **S**: both baselines right. **CW**: both predict the same winner but are wrong. **D**: baselines disagree. CW is *below* chance on MT-Bench—the collapse is not just “hard pairs.” D is above chance, suggesting a small real quality signal.

holds for all three. The inherited share rises slightly with more features. Length-only already gives 67–82% – so the numbers we report in the main paper are a lower bound.

Data	Judge	length	l+struct	full
Arena	mini	.724	.723	.740
Arena	4o	.673	.679	.703
MT-B	mini	.786	.839	.851
MT-B	4o	.806	.845	.878

Table 9: Inherited share under three feature sets (Qwen follows the same pattern). *length*: 6 features. *l+struct*: adds sentence, paragraph, and formatting features (13 total). *full*: all 16 symmetric features. The MT-Bench > Arena ranking is stable; numbers reported in the main paper (full) are a lower bound.

H Verbosity Experiments

This appendix details the intervention experiments of Section 4.5.

Sampling. We select MT-Bench decisive pairs where a majority of judges unanimously preferred the *shorter* response. This gives 198 pairs in which the initial verdict goes against length. The shorter response is preferred in only 22–27% of decisive pairs, so the 198 pairs cover nearly all such cases. In all of them, the losing response is longer than the winning response (median 260 vs. 68 tokens).

Padding. We use gpt-4o-2024-11-20 at temperature 0 to rewrite each losing response to roughly twice its original length. The model is instructed not to add information, and can only rephrase or expand existing sentences, restructure the response into bullets, or add summaries of existing content.

Verification. We use claude-haiku-4-5 to check each padded response for substantive additions. The verifier is from a different model family

than the three judges and checks for new facts, statistics, entities, examples, claims, or recommendations. Rephrasing, added length, and repeated summaries are not counted as additions. Of the 198 pairs, 133 pass this check and also exceed a $1.3\times$ length ratio. The remaining pairs are dropped. The verified pairs have a mean length ratio of $2.47\times$ (median $2.22\times$).

Re-judging. We re-judge each verified pair (original winner vs. padded loser) with all three judges in both display orders at temperature 0. Table 4 reports the position-averaged flip rate. Padding also increases position instability: 32–51% of pairs become position-inconsistent after padding, compared with 14–36% at baseline. These pairs are excluded when computing the flip rate.

I Per-Category Detail

Table 10 gives the per-category inheritance share underlying the pooled objective/subjective contrast of Section 4.6 (Table 5). We recompute the decomposition within each of MT-Bench’s eight categories, keeping the global out-of-fold baseline fixed and stratifying only the evaluation. We report reasoning and stem but exclude them from the objective/subjective pooling since neither fits cleanly into either group. The inherited share exceeds 76% in every objective category for all judges. extraction is lowest (62–78%). The 100% share for stem is based on only $n \approx 15$ residual examples. Given the small per-category sample sizes ($n \approx 80$ –120), we focus on the pooled contrast in the main text.

Category	Type	mini	4o	Qwen
math	obj	86.3	83.8	85.7
coding	obj	83.3	85.7	80.3
extraction	obj	61.9	76.5	78.3
reasoning	mix	78.3	82.0	87.0
stem	mix	96.1	100.0	100.0
writing	subj	89.7	92.3	91.9
roleplay	subj	83.8	84.3	84.7
humanities	subj	96.7	95.8	92.2

Table 10: Per-category inheritance share (%) on MT-Bench, with one column per judge (mini = gpt-4o-mini). *obj*: objective; *subj*: subjective; *mix*: reasoning and stem, reported but not pooled.

J Filtering Sensitivity and Funnel

Our main analysis drops human ties and position-inconsistent judge verdicts. Table 11 shows the

number of pairs removed at each step. The position-inconsistent rate is 11–18% for the GPT judges and 36% for Qwen2.5-7B on MT-Bench, making filtering more consequential for the open judge.

Data	Judge	Dec.	Drop _{pos}	Final
Arena	gpt-4o-mini	1000	105 (10.5%)	899
Arena	gpt-4o	1000	135 (13.5%)	905
Arena	Qwen2.5-7B	1000	306 (30.6%)	737
MT-B	gpt-4o-mini	927	134 (14.5%)	795
MT-B	gpt-4o	927	168 (18.1%)	816
MT-B	Qwen2.5-7B	927	337 (36.4%)	649

Table 11: Filtering funnel. Dec.: human-decisive pairs entering per-judge analysis. Drop_{pos}: pairs dropped for position inconsistency. Final: pairs decisive for both judge and human. MT-Bench starts from 927 pairs after dropping tied human votes (from 961 unique turn-1 pairs).

Table 12 tests whether the results depend on this filtering. We compare three regimes: *current*, which drops position-inconsistent pairs; *incl.*, which keeps them and treats each as a judge–human disagreement; and *AB*, which uses a single display order with no filtering. The agreement rate changes substantially across regimes. For Qwen2.5-7B on MT-Bench, for example, it drops from 0.84 under *current* to 0.60 under *incl.*. The inheritance share is more stable, changing by at most ≈ 8 pp across all cells.

Data	Judge	current	incl.	AB
Arena	gpt-4o-mini	74.0	74.2	73.1
Arena	gpt-4o	70.3	69.2	67.7
Arena	Qwen2.5-7B	73.4	68.5	66.1
MT-B	gpt-4o-mini	85.1	88.1	85.5
MT-B	gpt-4o	87.8	86.8	86.0
MT-B	Qwen2.5-7B	87.0	81.7	78.9

Table 12: Inheritance share (%) under three filtering regimes. *current*: main-paper policy. *incl.*: position-inconsistent pairs are kept and counted as judge–human disagreements. *AB*: single display order with no filtering.

The Flip-Test: A Behavioral Diagnostic for Multi-Agent Debate Failures

Julia Hu Alfred Shen Kumar Lakshmipathi

Amazon Web Services

{juliahu, alfreshe, laksku}@amazon.com

Abstract

Multi-agent debate has been proposed to improve language model reasoning through adversarial interaction, yet empirical studies report that debate underperforms simpler baselines on open-ended reasoning (Choi et al., 2025). We propose a behavioral diagnostic—the *flip-test*—that instruments debate at the example level and measures how often, and in which direction, agents change their answers between initial and final rounds. On 2,417 MuSiQue examples with each of Llama 3.1 8B and Ministral 3 8B, flips are common (65–70% of debates). Our central result conditions on *opportunity*: a flip destroys an available correct answer $\sim 2.4\times$ more often than it creates one (48.9% vs. 20.3% on Llama, 44.2% vs. 18.0% on Ministral; $p < 10^{-20}$). Unlike a raw directional split, this measure is invariant to how many correct answers were available, and it is statistically indistinguishable across the two model families. Three instruction-level interventions on a 100-example held-out set produce no improvement, suggesting simple prompt-level patches are insufficient. We separate this behavioral finding from mechanism: direct opponent-adoption is low ($\sim 10\%$), so the asymmetry is consistent with sycophancy but does not establish it. The signature is also scale-dependent—on three 70B-class models the flip rate roughly halves and debate turns net-positive—placing the harmful regime in the 8B capacity band rather than in debate itself. We frame the flip-test as a cheap behavioural probe for locating that regime.

1 Introduction

Multi-agent debate has attracted research interest as an inference-time mechanism for improving large language model (LLM) reasoning (Irv-ing et al., 2018; Du et al., 2024; Liang et al., 2024). The motivating intuition is that agents producing diverse initial answers and critiquing one another should propagate evidence and converge

on the correct answer more often than a single chain of reasoning. Recent empirical work complicates this picture: Choi et al. (2025) report that majority voting accounts for most of the gain attributed to debate, and adaptive routing (Eo et al., 2025) confirms debate is harmful on a substantial fraction of examples.

We address a specific gap: a *behavioral* description of where debate fails. We instrument the debate process and record, on each example, whether the endorsed answer moved toward correctness or away from it. We define a *flip* operationally—an agent’s initial answer differs from the moderator’s final answer—and, rather than scoring the raw direction of flips (which a difference in base rates can bias), we condition on whether a correct answer was available to lose, giving a low-cost diagnostic that assumes no particular causal mechanism.

Contributions. (1) The flip-test: a per-example diagnostic that decomposes debate behavior into a flip rate and an *opportunity-matched* destroy-vs-create comparison, the latter designed to avoid the base-rate bias that inflates or deflates raw directional counts. (2) On 2,417 MuSiQue examples per model, conditional on a flip debate destroys an available correct answer $\sim 2.4\times$ more readily than it creates one ($p < 10^{-20}$ on both), an effect of equal magnitude across two independently developed model families (+28.6 vs. +26.2 pp). (3) A held-out study ($n = 100$) finds three instruction-level interventions all at or below baseline, indicating prompt-level patches are insufficient. (4) The signature reverses at 70B scale (§4.5), locating the harmful regime in the 8B capacity band. Direct opponent-adoption is low ($\sim 10\%$), so we read the asymmetry as consistent with sycophancy, not a proof of it.

2 Related Work

Multi-agent debate. Du et al. (2024) report that debate improves factual accuracy when initial answers are diverse and incorrect in different ways. Liang et al. (2024) introduce a divergence-encouraging variant after observing that LLMs “lock in” to a single solution. These works document conditions under which debate succeeds; we examine why it fails.

Voting and self-consistency. Wang et al. (2023) show that sampling reasoning paths and taking the majority recovers most prompting gains. Choi et al. (2025) extend this to debate: voting accounts for “most of the performance gains attributed to multi-agent debate.” Our diagnostic complements this finding by showing that debate’s marginal contribution is limited by frequent, accuracy-reducing answer changes—a pattern consistent with, but not proof of, sycophantic interaction.

Adaptive routing. Eo et al. (2025) activate debate only when a cheaper baseline is uncertain. Their finding that most examples do not benefit from debate aligns with our observation that flips are net-harmful; we explain *why* the harmful examples exist.

Sycophancy. Sharma et al. (2024) show LLMs systematically defer to user assertions, including incorrect ones, and that RLHF reinforces this. Our flip asymmetry is one of several findings that would be consistent with sycophancy operating in agent-to-agent interaction; we discuss in §5.2 why the present data do not establish this directly.

Concurrent analyses of debate failure. A parallel line of recent work dissects debate breakdowns. Hao et al. (2026) decompose stance convergence and argue that not all answer changes reflect conformity, distinguishing principled revision from social pressure. Wynn et al. (2025) catalogue failure modes and show that inter-agent communication can actively degrade performance. Bertalanič and Fortuna (2026) find that isolated self-correction can outperform unguided homogeneous debate. Our contribution is complementary and methodological rather than taxonomic: where these works characterize *what* goes wrong, the flip-test supplies a cheap, per-example *instrument*—a flip rate plus an opportunity-matched destroy-vs-create rate that any practitioner can

compute on their own workload without ground-truth-labeled failure categories, and which we use here to expose a scale-dependent reversal (§4.5) that a purely qualitative taxonomy would miss. We also contribute the negative result that instruction-level patches do not close the gap (§4), consistent with Bertalanič and Fortuna (2026)’s finding that guidance from within the debate is limited, and with evidence that LLMs struggle to self-correct reasoning without external signal (Huang et al., 2024; Madaan et al., 2023).

3 Methods

Datasets. We evaluate on the MuSiQue-Ans development set (Trivedi et al., 2022), treating all 2,417 multi-hop questions as our primary cell, and on GSM8K (Cobbe et al., 2021) (1,319 test items) as a consistency check. Correctness on MuSiQue is token F1 ≥ 0.5 ; on GSM8K it is exact numeric match.

Models. Two instruction-tuned 8B-class models served via Amazon Bedrock: Llama 3.1 8B Instruct (Meta) and Ministral 3 8B Instruct (Mistral AI). They differ in pretraining corpus, RLHF pipeline, and developer; they are not chosen as a scaling axis.

Protocols. **Greedy** (1 call): temp 0, 960 tokens. **Vote3** (3 calls): temp 0.7, 320 tokens each, majority vote. **Debate** (7 calls): two agents at temp 0.7 generate initial answers (192 tokens each), exchange a rebuttal round (128 each), exchange a closing round (64 each), and a temp-0 moderator (192 tokens) returns the final answer. All protocols share a 960-token ceiling; realized medians are ~ 100 (greedy), ~ 160 (vote3), ~ 500 (debate).

Flip definition. For each agent $a \in \{A, B\}$, we compare the agent’s initial round-1 answer $a_0(x)$ to the moderator’s final answer $a_f(x)$ (lowercased, punctuation-stripped, whitespace-collapsed exact match). An example is *flipped* if at least one agent flips. Because the moderator’s output is the protocol’s actionable answer, this is an *endorsement-level* flip—the protocol stops endorsing an agent’s initial answer—rather than a literal within-agent self-revision (§5.2). A flipped example is correct $\rightarrow$ wrong if at least one initial answer matched gold and the final is wrong, wrong $\rightarrow$ correct if no initial matched but the final is correct, and otherwise “stay-class.” Directional percentages are over correct $\rightarrow$ wrong $\cup$

wrong→correct only.

Statistics. For each model we report the no-flip vs. flip accuracy gap with a bootstrap 95% CI (10,000 resamples, seed 42). Our primary directional statistic is *opportunity-matched* (below); we report the raw 50/50 binomial only as a descriptive reference.

Why the raw 50/50 null is the wrong test. A correct→wrong flip (“destroy”) is possible only when ≥ 1 initial answer is correct; a wrong→correct flip (“create”) only when neither is. These draw from different conditioning sets, so a 50/50 null is one the design already violates—and the base rate runs *with* the asymmetry: for two agents at accuracy p , a correct answer is available with probability $1 - (1 - p)^2 > 0.5$ whenever $p > 29.3\%$, well below the observed greedy 40.3%/55.6%. The raw majority could thus be a pure opportunity artifact. We instead compare the *destroy rate* $\Pr(\text{final wrong} \mid \geq 1 \text{ correct, flipped})$ to the *create rate* $\Pr(\text{final correct} \mid \text{none correct, flipped})$; their ratio and a two-proportion test are invariant to how many correct answers were available.

Caveat: no-flip $\equiv$ initial agreement by construction. Under exact match, a no-flip example is one where neither initial answer differs from the final, which entails the two initials agree; so the flip rate is essentially one minus the initial-disagreement rate. (Empirically 816/840 and 705/718 rather than the definitional 840/840/718/718; the few exceptions are normalization mismatches, not genuine agree-then-flip events.) Hence the no-flip/flip accuracy gap mostly re-describes that agreement correlates with easiness (Du et al., 2024); we treat it as descriptive. This is exactly why our headline conditions on *opportunity*, not flip status—the destroy-vs-create comparison lives *within* the flipped set, holding correct-answer availability fixed.

4 Results

4.1 Flips correlate with accuracy loss

Figure 1 reports the descriptive gap. On Llama, debate accuracy is 66.1% [62.8, 69.2] without a flip and 31.0% [28.8, 33.3] with a flip; Ministral: 75.3% [72.1, 78.4] vs. 33.7% [31.5, 36.0]. Because no-flip is initial agreement by construction (§3), we read this gap as *easy-vs-hard* rather

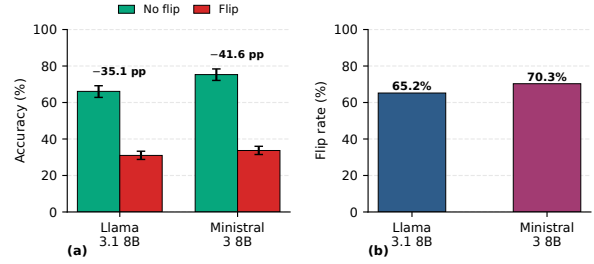


Figure 1: (a) Accuracy conditional on whether at least one agent flipped its answer during debate. Error bars are bootstrap 95% CIs over the two (disjoint) example groups. Because no-flip $\equiv$ initial agreement (§3), this gap is descriptive—it contrasts easy agreed examples with harder disagreed ones—and is *not* our causal headline; see §4.2. (b) Flip rate on MuSiQue: 65.2% (Llama) and 70.3% (Ministral).

than as a causal effect of suppressing flips, and we do not rest any mechanistic claim on it. Notably, most of the cross-model difference lives in the no-flip cell (75.3 vs. 66.1, +9.2 pp) while the flip-conditional accuracies are nearly equal (33.7 vs. 31.0), i.e. the two models differ mainly in baseline ability on easy items, not in how badly flips hurt. The causal headline is instead the opportunity-matched result in §4.2.

4.2 Conditional on a flip, debate destroys correct answers $\sim 2.4\times$ more than it creates them

We compare the destroy and create rates defined in §3 (Figure 2). When a flip has a correct answer to lose, debate loses it $269/550 = 48.9\%$ of the time on Llama; when it has none to gain, it gains one only $208/1026 = 20.3\%$ of the time—a $2.41\times$ ratio. Ministral is nearly identical: 44.2% (313/708) vs. 18.0% (178/991), $2.46\times$. A two-proportion test separates the two rates decisively on both models ($z \approx 11.8$, $p < 10^{-20}$): *conditional on changing its endorsed answer, debate is $\sim 2.4\times$ more likely to abandon a correct answer than to reach one*. The gap is the same size on both families (destroy–create = +28.6 vs. +26.2 pp), so it is a shared property, not a quirk of one model.

The non-directional majority. Directional flips are the minority: $\sim 70\%$ of flips are *stay-class* (the endorsed answer changes string but not correctness). These carry no accuracy signal—hence their exclusion from the destroy/create rates—but their prevalence shows most endorsement changes are churn, not error correction, driving debate’s to-

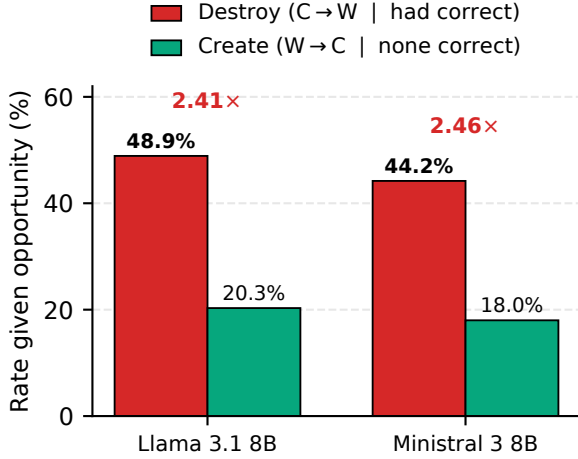


Figure 2: Opportunity-matched flip outcomes. *Destroy rate* = $\Pr(\text{final wrong} \mid \geq 1 \text{ initial correct})$; *create rate* = $\Pr(\text{final correct} \mid \text{no initial correct})$, among flipped examples. Debate destroys an available correct answer $\sim 2.4\times$ more readily than it manufactures one on both models ($p < 10^{-20}$, two-proportion test). Once matched for opportunity the two models are statistically indistinguishable (+28.6 vs. +26.2 pp), i.e. Ministral’s larger *raw* lean is just its having more correct answers to lose.

ken cost without accuracy return (§4.4).

4.3 Instruction-level interventions: a small held-out study

To probe whether prompt-level patches can reduce flipping, we ran a small held-out study on $n = 100$ MuSiQue examples with Llama 3.1 8B. This held-out subset has a baseline debate accuracy of 40%, slightly below the full-set 43.2% reported in Figure 3(a); the two figures refer to different sample sizes ($n = 100$ vs. $N = 2,417$) and should not be expected to coincide. All intervention deltas below are relative to the $n = 100$ baseline. **Anchor-ing** added: “Only change your answer if you identify a fundamental flaw in your original reasoning” (40% $\rightarrow$ 40%). **Justification-on-flip** required each agent to articulate the specific flaw before changing answers (40% $\rightarrow$ 39%). **Moderator review** asked the moderator to reject flips that lacked substantive justification (40% $\rightarrow$ 38%). None of the three reached baseline. With $n = 100$, individual differences are within sampling noise; the pattern that all three stayed at or below baseline is suggestive, not definitive, and a stronger conclusion would require broader prompt-engineering studies.

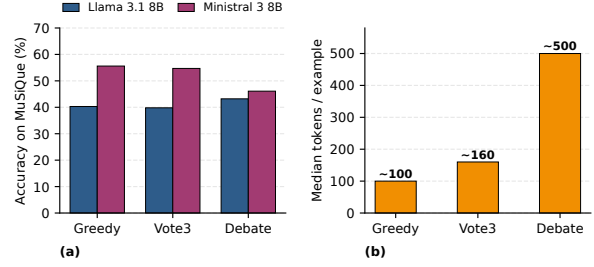


Figure 3: (a) Protocol accuracy on MuSiQue. Debate beats greedy on Llama by 2.9 pp but loses to greedy on Ministral by 9.5 pp; debate is not a uniformly best fixed protocol. (b) Median tokens generated per example. Debate consumes roughly $5\times$ greedy and $3\times$ vote3.

4.4 Protocol comparison and cost

Debate is dominant on Llama-MuSiQue (43.2% vs. greedy 40.3%) but is dominated by greedy on Ministral-MuSiQue (46.1% vs. 55.6%, Figure 3a). This dataset/model dependence rules out treating debate as a universal default. Crucially, debate’s token cost is roughly $5\times$ greedy and $3\times$ vote3 (Figure 3b), so any small accuracy gain comes at substantial expense.

GSM8K consistency check. The flip-test reproduces qualitatively on GSM8K: flip-conditional accuracy is far below no-flip (Llama: 49.5% vs. 88.1%; Ministral: 57.6% vs. 93.4%) and directional flips lean correct $\rightarrow$ wrong on both models ($\sim 77\%$). We do not foreground GSM8K because arithmetic and multi-hop QA differ in normalization and ambiguity, but the pattern is consistent.

4.5 The flip signature reverses at 70B scale

The most common question about an 8B-only result is whether it is an artifact of capacity (§5.1). To probe this directly, we ran the identical MuSiQue protocol on three 70B-class models—Llama 3.1 70B, Llama 3.3 70B, and Mistral Large—and applied the same flip-test.

Two things change together at scale (Table 1). The flip rate roughly halves (65–70% $\rightarrow$ 29–40%), and the sign of debate’s net effect flips: debate *beats* both baselines on all three 70B models (+6.9, +2.5, +1.5 pp over greedy), whereas at 8B it hurt Ministral by 9.5 pp. This is direct evidence that the harmful-flip regime is a property of the 8B capacity band, not of debate in general, favoring the non-monotonic account of §5.1. We do *not* over-claim a predictive rule from five cells: flip rate vs. the debate–greedy margin is $r = -0.62$ ($p = 0.26$, $n = 5$), so flip rate is a weak screen,

Model	Greedy	Vote3	Debate	Flip rate
<i>8B-class (primary)</i>				
Llama 3.1 8B	40.3	39.8	43.2	65.2%
Minstral 3 8B	55.6	54.7	46.1	70.3%
<i>70B-class</i>				
Llama 3.1 70B	57.2	56.7	64.1	39.0%
Llama 3.3 70B	60.8	61.7	63.3	39.9%
Mistral Large	46.7	46.6	48.2	29.2%

Table 1: Accuracy (%) and flip rate on MuSiQue ($N = 2,417$); best protocol per row in bold. Vote3 is included because Choi et al. (2025) attribute most of debate’s gain to voting and debate costs $\sim 3 \times$ Vote3’s tokens—yet debate never beats Vote3 by more than ~ 2.5 pp, and at 8B Vote3 essentially ties Greedy. At 70B scale debate beats both baselines on all three models and the flip rate roughly halves, reversing the harmful 8B pattern.

not a decision procedure. (Accuracy and flip rate here reproduce exactly against the 8B cells; a harmonized directional breakdown at 70B is deferred to avoid mixing correctness conventions.)

5 Discussion

5.1 Are flips just because the models are small?

One explanation is the *weak-reasoning hypothesis*: 8B models are too weak to tell a correct answer from a plausible wrong one, so they oscillate at random. Random oscillation predicts equal destroy and create rates; the $\sim 2.4 \times$ asymmetry (§4.2) rules that out. What remains open is a *non-monotonic* capacity story: mid-capacity models may sense a shaky correct answer without being able to defend it, while stronger models leave the regime entirely. The 70B results (§4.5) fit that exit—flips grow rare and net-positive—but two 8B points cannot separate a genuine mid-capacity peak from a smooth capacity trend, so we do not claim one.

5.2 What the asymmetry does and does not show

A natural interpretation is sycophantic capitulation (Sharma et al., 2024): agents abandon correct answers under confident wrong opposition. To probe this we measured the *adoption rate*—among initially-disagreeing examples, how often the final wrong answer normalize-equals the opponent’s initial wrong answer. It is only $\sim 10\%$ on both models; the remaining correct→wrong finals are paraphrases or new wrong formulations. So

the asymmetry is real but does not establish literal answer-adoption; the mechanism may instead be moderator synthesis bias, or initially-correct positions being less defensible on hard instances.

Endorsement-level, and what it does not isolate.

A flip is defined at the *endorsement* level (initial answer differs from the moderator’s final output), not as a within-agent revision. This mixes three processes the present logs cannot separate: (i) peer persuasion, (ii) moderator synthesis bias, and (iii) spontaneous instability a model would show under *any* re-prompting. Two absent controls would disentangle them—a *self-reflection* control (one agent reconsiders, no peer) bounding (iii), and a *moderator-free* variant (agents self-finalize) separating (i) from (ii). Lacking both, our rates are *upper bounds* on peer-induced change; adding these ablations is the highest-value next experiment.

5.3 Limitations and recommendation

Our study covers five models across two capacity tiers, one debate topology, and primarily MuSiQue; the intervention study is small ($n=100$); and the causal mechanism behind the asymmetry remains open (§5.2). We therefore position the flip-test as a behavioural instrument, not a protocol-selection rule: with only five cells the flip-rate/margin correlation is not significant, so we make no prescriptive claim. What the diagnostic does deliver is a cheap, opportunity-matched readout of whether debate is in its harmful regime on a given model, and evidence that this regime is real at 8B and recedes by 70B.

References

- Blaž Bertalanč and Carolina Fortuna. 2026. [The cost of consensus: Isolated self-correction prevails over unguided homogeneous multi-agent debate](#). *arXiv preprint arXiv:2605.00914*.
- Hyeong Kyu Choi, Jerry Zhu, and Sharon Li. 2025. Debate or vote: Which yields better decisions in multi-agent large language models? In *NeurIPS*. Spotlight.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving

- factuality and reasoning in language models through multiagent debate. In *ICML*.
- Sugyeong Eo, Hyeonseok Moon, Evelyn Hayoon Zi, Chanjun Park, and Heuiseok Lim. 2025. [Debate only when necessary: Adaptive multiagent collaboration for efficient LLM reasoning](#). *arXiv preprint arXiv:2504.05047*.
- Xiqi Hao, Zengqing Wu, Yu-Xuan Qiu, Chuan Xiao, Ruiqi Xu, Shuyuan Zheng, and Jianbin Qin. 2026. [Not all flips are conformity: Decomposing stance convergence in multi-agent LLM debate](#). *arXiv preprint arXiv:2606.00820*.
- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. [Large language models cannot self-correct reasoning yet](#). In *ICLR*.
- Geoffrey Irving, Paul Christiano, and Dario Amodei. 2018. [AI safety via debate](#). *arXiv preprint arXiv:1805.00899*.
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. [Encouraging divergent thinking in large language models through multi-agent debate](#). In *EMNLP*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#). In *NeurIPS*.
- Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. 2024. [Towards understanding sycophancy in language models](#). In *ICLR*.
- Harsh Trivedi, Niranjana Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop questions via single-hop question composition. In *TACL*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *ICLR*.
- Andrea Wynn, Harsh Satija, and Gillian Hadfield. 2025. [Talk isn’t always cheap: Understanding failure modes in multi-agent debate](#). *arXiv preprint arXiv:2509.05396*.

Activation-Based Active Learning for In-Context Learning: Challenges and Insights

Yaseen M. Osman, Geoff V. Merrett, Stuart E. Middleton

University of Southampton, UK

y.m.osman@soton.ac.uk, gvm@ecs.soton.ac.uk, sem03@soton.ac.uk

Abstract

Deep active learning has previously been explored for LLM in-context sample selection, but not with methods that utilise recent advances in understanding of transformer activations. In this paper, we test the hypothesis that model activations could provide a fine-grained signal to optimise the selection of in-context examples. We present a comprehensive analysis of MLP activation-based deep active learning methods applied to in-context learning, including how different attention masking strategies impact active learning across diverse classification and generative datasets, using both Llama-3.2-3B and Qwen2.5-3B base models. However, we find a negative result: MLP and embedding layer outputs, viewed through the lenses of massive activations or the first four moments, do not correlate with example quality or task performance. Specifically, the absolute Spearman correlation coefficient is at most 0.33 for all tasks and models we tested, showing that such activation-based sampling should not be used for in-context learning. We hypothesise that this may be due to superposition, whereby models represent more features than they have dimensionality, suggesting that methods like Sparse Autoencoders (SAEs) may be a promising future direction.¹²

1 Introduction

In-context learning is a popular alternative to fine-tuning for improving the performance of Large Language Models (LLMs). It is the process of providing examples to the LLM in the prompt rather than updating the model weights (Brown et al., 2020). Recent studies have looked at the impact and sensitivity of LLMs towards the provided in-context examples (Schoch and Ji, 2025; Li and Qiu,

2023; Ye et al., 2023; Qin et al., 2024; Rubin et al., 2022; Zhang et al., 2022; Yao et al., 2024; Wu et al., 2023).

Active learning (Settles, 2009; Cohn et al., 1994; Ein-Dor et al., 2020) aims to optimise the data selection phase by selecting data that is likely to yield better model training performance. Active learning has been extensively studied for machine learning models, such as Support Vector Machines (SVMs) (Tong, 2001) and neural networks (Cohn et al., 1994). Recently, deep active learning has been used with deep learning models such as transformers (Liu et al., 2022b; Zhang and Zhang, 2020; Shelmanov et al., 2019; Ein-Dor et al., 2020; Bayer et al., 2026).

Deep active learning has been also adapted for sampling in-context learning examples for LLMs through a variety of techniques, such as using model uncertainty (Margatina et al., 2023; Wang et al., 2025), data diversity (Margatina et al., 2023; Li and Qiu, 2023; Lee et al., 2025; Malik et al., 2025), mixtures of both diversity and uncertainty (Mavromatis et al., 2023; Treerath and Pittorino, 2026), similarity between in-context examples and test samples encoded through sentence encoders (Liu et al., 2022a; Margatina et al., 2023), training a specialised retrieval-based model (Luo et al., 2023), Monte Carlo sampling (Schoch and Ji, 2025) and reinforcement learning (Zhang et al., 2022).

The hypothesis of this paper is that model activations could provide a fine-grained signal to optimise the selection of in-context examples, grounded in both the transformer layer weights, which encode LLM pre-trained knowledge, and the relevance of the in-context example to the test sample. In this paper, we focus on MLP layer activations given the crucial role of the MLP layer in knowledge memorisation (Dong et al., 2025; Bricken et al., 2023; Nanda et al., 2023). We also explore adjusting causal masking (BehnamGhader et al., 2024; Lin et al., 2025) to better capture com-

¹Our code and a link to our project page containing more results are available at <https://github.com/yaseenosman/activation-based-al>.

²An arXiv.org version of this work is available at <https://arxiv.org/pdf/2606.05134>.

plex relationships and enable greater information flow between the candidate in-context learning example and test sample.

The contribution in this paper is a comprehensive analysis of MLP activation-based active learning methods applied to in-context learning to date, covering a diverse range of datasets comprising both classification and generative tasks. We include a study examining how different LLM masking strategies impact active learning, including tailored masking methods for in-context learning sampling. Our results show that activation-based methods for active learning do not correlate directly with task performance. This is a negative result for our hypothesis. However, this study provides a contribution in terms of useful insights and challenges that we derive from our results to guide future research in this area.

2 Related Work

Attention Masking Adjustments Recent work explored removing the causal masking from pre-trained decoder-only transformer models to enhance their performance in encoder-based and embedding tasks by allowing the model to process full sequences bidirectionally (BehnamGhader et al., 2024; Lin et al., 2025). Additionally, Springer et al. (2025) and Leviathan et al. (2025) achieved a similar effect by instead duplicating the prompt twice in the input. Other work focusing on studying and adjusting attention masking includes Barbero et al. (2024, 2025); Swietojanski et al. (2023); Wu et al. (2024).

In-Context Data Sampling Adjacent, for in-context learning examples sampling, Liu et al. (2022a) and Margatina et al. (2023) have utilised another model’s embeddings to encode both examples and tests, then employed the k-nearest neighbours algorithm. Furthermore, sampling using the attention weights of the queried transformer model has been explored in the context of Retrieval-Augmented Generation (RAG) (Chen et al., 2025) and Visual Instruction Tuning (Hu et al., 2025). We consider these sampling techniques to be the most relevant.

Mechanistic Interpretability of Activations and In-Context Learning Several works have used Sparse Autoencoders (SAEs) to facilitate the interpretation of activations and mitigate difficulties from superposition (Sharkey et al., 2022; Bricken

et al., 2023; Huben et al., 2024). Superposition is when the model represents more features than its dimensionality, resulting in highly uninterpretable and entangled activation matrices (Elhage et al., 2022).

Notably, Cho and Hockenmaier (2025) used SAEs to perform downstream steering to increase the in-context learning performance of a model. On the other hand, Cho et al. (2025) utilised SAEs beyond interpretability and found that SAEs reconstruction loss correlates with prompt task performance for in-context learning settings.

Moreover, Olsson et al. (2022) attributed the emergence of the in-context learning ability to specialised attention heads called induction heads.

3 Method

In this section, we present our methodology for scoring in-context examples using active learning and how different types of masking impact sampling performance.

3.1 Masking

To allow for different information-flow variants between the example and the test sample, we explored various masks. Given a 1-shot prompt, these are:

- **Causal Mask**, which blocks the attention from future tokens (i.e., default masking).
- **No Mask**, which removes the causal masking following (Lin et al., 2025).
- **Row Test-Based**, which only masks the test *queries* (i.e., rows of test tokens in attention weights).
- **Col Test-Based**, which only masks the test *keys* (i.e., columns of test tokens in attention weights).

Figure 1 depicts these maskings.

3.2 In-Context Examples Scoring

Once the masking of a model, containing L layers, has been adjusted, given a set of candidate examples M and test samples N , for each example $m \in M$, we construct $len(N)$ 1-shot prompts. Specifically, for an $n \in N$, $p_{m,n}$ is the concatenation of example m , a delimiter " $\text{\textbackslash n\textbackslash n}$ ", n . We then input this as a prompt to the model for a single forward pass, resulting in $L + 1$ output hidden states as activation matrices (i.e., one for each MLP layer

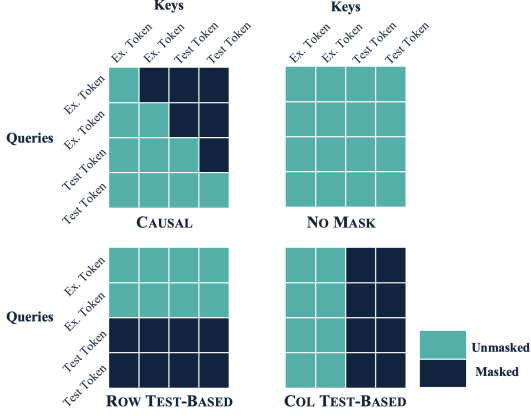


Figure 1: Shows a visualisation of the different attention maskings used in this paper using a prompt consisting of both example tokens (i.e., Ex. Token) and test tokens.

and for the embedding layer) $\{X_0, \dots, X_L\}$. The score of the prompt is given by:

$$pt_score(p_{m,n}) = \frac{1}{L+1} \sum_{l=0}^L metric(X_l), \quad (1)$$

such that $metric()$ is a metric of choice to reduce an activation matrix to a score.

To get the overall score of m , we take the mean of all prompt scores that consist of m , as follows:

$$st_score(m) = \frac{1}{len(N)} \sum_{n \in N} pt_score(p_{m,n}). \quad (2)$$

3.3 Metric Choices

The choices of metrics used in Equation 1, given X_l , are the first four moments: mean, variance, skewness, and kurtosis. These moments allow the capturing of various activation patterns. For the mean, we take the absolute value of X_l before calculating its mean. This is because we empirically observe that the mean of MLP activations is close to zero, owing to the presence of many positive and negative activation values. We do not make any changes to the other three moments.

We use the largest massive activation, given its importance in model performance (Sun et al., 2024), as an additional metric by selecting the activation with the largest absolute magnitude in X_l . We refer to this as the max activation. For completeness, we also explore the max activation, but for each feature vector and take the mean across all vectors in X_l . We refer to it as per-vector max.

4 Experimental Setup

In this section, we outline our experimental setup in detail before presenting our results. We conduct all our evaluations using the LM evaluation harness framework (Gao et al., 2023).

4.1 Models

We mainly experimented with two models, which are Llama-3.2-3B (Grattafiori et al., 2024), and Qwen2.5-3B (Team, 2024; Yang et al., 2024). These provide examples for two different families that are frequently used. We also explored Llama-3.1-8B and Llama3.1-8B-Instruct.

4.2 Datasets

We use a diverse set of datasets, covering classification-based (i.e., two multiple-choice tasks and a true/false task) and generative-based tasks (i.e., a math reasoning task). These are ARC-Challenge (ARC-C) (Clark et al., 2018), OpenBookQA (OBQA) (Mihaylov et al., 2018), BoolQ (Clark et al., 2019; Wang et al., 2019), and GSM8K (Cobbe et al., 2021), with the number of test samples, $len(N)$, being 1172, 500, 3270, and 1319, respectively.

4.3 Sampling and Evaluation

For each dataset, we randomly sample a subset M of 1000 in-context example candidates from its training split. However, for ARC-C, we use the full training split of 1119 samples as it is close to 1000.

Moreover, we calculate the st_score for all examples in M . We also evaluate each of the resulting $p_{m,n}$ 1-shot prompts on the task. Specifically, we evaluate task performance using exact match with strict match filtering for GSM8K and using accuracy for the other datasets. Finally, we compute the Spearman correlation coefficient between the task performance and st_score through utilising the SciPy implementation (i.e., `scipy.stats.spearmanr`) (Virtanen et al., 2020).

5 Results

We focus solely on 1-shot in-context learning to simplify the exploration and reduce the computational cost of our experiment, thereby enabling a finer-grained analysis. However, our scoring method can be applied to few-shot settings. Lastly, we use a batch size of 1 for all our experiments on an Nvidia L4, except for GSM8K, which we run

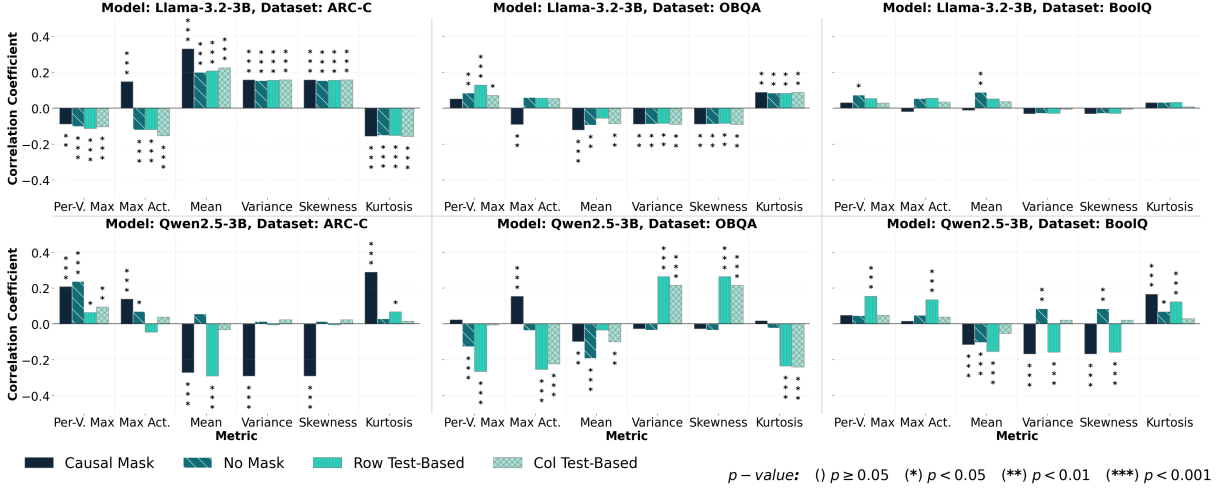


Figure 2: Shows the Spearman correlation and the associated p -values between task performance and the different activation-based metrics discussed. Results are for both Llama-3.2-3B and Qwen2.5-3B across three different classification-based datasets.

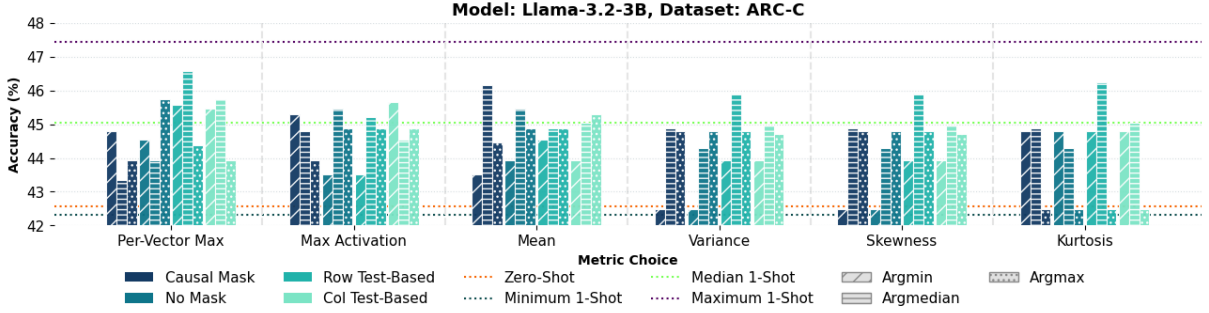


Figure 3: Shows the accuracy of Llama-3.2-3B on ARC-C when sampling examples with argmin, argmedian, and argmax metric values (column pattern) using the different masks (column colour). The accuracy does not align with argmin, argmedian, and argmax. The performances of the examples of the maximum, median and minimum accuracies are also included.

on an Nvidia L40 with a batch size of 64 due to the computational time of the generative task.

In Figure 2, we present the correlation coefficient across three classification-based datasets. For both Qwen2.5-3B and Llama-3.2-3B, a consistently low correlation coefficient is observed. Additionally, low p -values (i.e., $p < 0.001$) are observed across several experiments, reflecting a lack of correlation with high statistical significance. However, some results, especially when using Qwen2.5-3B or both models on BoolQ, have relatively higher p -values (i.e., $p \geq 0.05$), indicating lower statistical significance. We argue that the lack of correlation is persistent due to the very low correlation values in these results, which approach zero (i.e., < 0.1). Furthermore, the highest overall correlation value achieved is only around 0.33.

Figure 3 includes the performance of examples that achieve the maximum, median, and minimum

scores under each metric and mask combination on ARC-C using Llama-3.2-B, referred to as argmax, argmedian, and argmin, respectively. The true best example is never selected. Moreover, across all metrics and masks, performance and score magnitude do not align. Additionally, argmax and argmin values indicate high variability in 1-shot performance, and some examples drop performance even below the zero-shot accuracy, aligning with previous work, such as Schoch and Ji (2025).

We expanded our Llama-3.2-3B to GSM8K example reasoning-based generative tasks in Figure 4. However, the lack of correlation persists across all masks. The NaN value resulting from using causal masking with the max activation metric is due to the largest massive activation value not changing across all prompts. This results in zero variance, which in turn results in an undefined Spearman correlation.

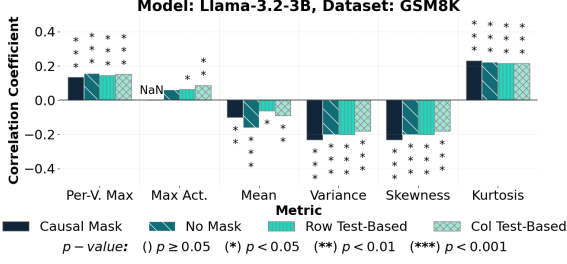


Figure 4: Shows the Spearman correlation and p -values between task performance and different activation-based metrics for GSM8K using Llama-3.2-3B.

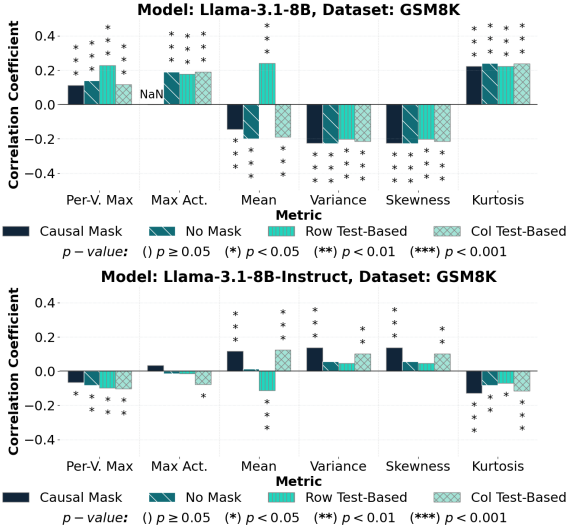


Figure 5: Shows the Spearman correlation and p -values between task performance and different activation-based metrics for GSM8K using Llama-3.1-8B and Llama-3.1-8B-Instruct.

Additionally, we run our GSM8K experiments on both Llama-3.1-8B and Llama-3.1-8B-Instruct to cover a wider variety of model sizes and variants. The observation of no correlation is still maintained, as shown in Figure 5. The absence of the NaN value in the case of the instruct variant (i.e., Llama-3.1-8B-Instruct) when computing the max activation metric for the causal mask is due to extremely low variance (i.e., around $7e-9$) in the example scores. This is in contrast to the zero variance in the cases of Llama-3.1-8B and Llama-3.2-3B.

Based on the above, we conclude that there is no reliable correlation between activation patterns and in-context example performance across the metrics we have explored. We hypothesise that this is due to the superposition phenomenon, resulting in each feature dimension representing complex, interconnected features between dimensions, requiring a non-linear filtering approach (Elhage et al.,

2022). Hence, we highlight the use of Sparse Autoencoders (SAEs) (Huben et al., 2024; Cho et al., 2025) being an alternative to the metrics used in this paper, as a promising future direction to advance further.

6 Conclusion

In this work, we showed the significantly low correlation between task performance and in-context example scores using MLP and embedding layer activations (i.e., per-layer output hidden states) across a diverse set of tasks. Thus, we argue that MLP activations should not be used to sample in-context examples without further disentangling the MLP, and that practitioners should use other sampling methods that yield more reliable scores (e.g., model uncertainty or attention weights). We hope these findings contribute to both LLMs’ explainability and active learning efforts.

Limitations

Due to computational costs and to provide finer-grained analysis, we examine only 1-shot prompts in this paper. We expect results to hold for higher shot counts, but leave it for future work to confirm this.

Acknowledgments

This work was supported by the UK Research and Innovation Centre for Doctoral Training in Machine Intelligence for Nano-electronic Devices and Systems [EP/S024298/1]. The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

References

- Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Petar Veličković, Razvan Pascanu, and Michael M. Bronstein. 2025. [Why do LLMs attend to the first token?](#) In *Second Conference on Language Modeling*.
- Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, João G.M. Araújo, Alex Vitvitskyi, Razvan Pascanu, and Petar Veličković. 2024. [Transformers need glasses! information oversquashing in language tasks](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 98111–98142. Curran Associates, Inc.

- Markus Bayer, Justin Lutz, and Christian Reuter. 2026. [ActiveLLM: Large language model-based active learning for textual few-shot scenarios](#). *Transactions of the Association for Computational Linguistics*, 14:1–22.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. [LLM2vec: Large language models are secretly powerful text encoders](#). In *First Conference on Language Modeling*.
- Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermy, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. [Towards monosemanticity: Decomposing language models with dictionary learning](#). *Transformer Circuits Thread*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Shijie Chen, Bernal Jimenez Gutierrez, and Yu Su. 2025. [Attention in large language models yields efficient zero-shot re-rankers](#). In *International Conference on Learning Representations*, volume 2025, pages 73047–73069.
- Ikhyun Cho and Julia Hockenmaier. 2025. [Toward efficient sparse autoencoder-guided steering for improved in-context learning in large language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28961–28973, Suzhou, China. Association for Computational Linguistics.
- Ikhyun Cho, Gaeul Kwon, and Julia Hockenmaier. 2025. [On the versatility of sparse autoencoders for in-context learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19531–19538, Suzhou, China. Association for Computational Linguistics.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. [BoolQ: Exploring the surprising difficulty of natural yes/no questions](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- David Cohn, Zoubin Ghahramani, and Michael Jordan. 1994. [Active learning with statistical models](#). In *Advances in Neural Information Processing Systems*, volume 7. MIT Press.
- Yihe Dong, Lorenzo Noci, Mikhail Khodak, and Mufan Li. 2025. Is random attention sufficient for sequence modeling? disentangling trainable components in the transformer. *arXiv preprint arXiv:2506.01115*.
- Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. [Active Learning for BERT: An Empirical Study](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.
- Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. [Toy models of superposition](#). *Transformer Circuits Thread*.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2023. [A framework for few-shot language model evaluation](#).
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Jucheng Hu, Suorong Yang, and Dongzhan Zhou. 2025. δ -attnmask: Attention-guided masked hidden states for efficient data selection and augmentation. *arXiv preprint arXiv:2508.09199*.
- Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. [Sparse autoencoders find highly interpretable features in language models](#). In *The Twelfth International Conference on Learning Representations*.
- Jiho Lee, Kyungjae Lee, and Eunwoo Kim. 2025. [Contrastive in-context learning with active memory for](#)

- [task planning](#). In *Workshop on Making Sense of Data in Robotics: Composition, Curation, and Interpretability at Scale at CoRL 2025*.
- Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2025. Prompt repetition improves non-reasoning llms. *arXiv preprint arXiv:2512.14982*.
- Xiaonan Li and Xipeng Qiu. 2023. [Finding support examples for in-context learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.
- Ziyong Lin, Haoyi Wu, Shu Wang, Kewei Tu, Zilong Zheng, and Zixia Jia. 2025. [Look both ways and no sink: Converting LLMs into text encoders without training](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22839–22853, Vienna, Austria. Association for Computational Linguistics.
- Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022a. [What makes good in-context examples for GPT-3?](#) In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.
- Mingyi Liu, Zhiying Tu, Tong Zhang, Tonghua Su, Xiaofei Xu, and Zhongjie Wang. 2022b. [Ltp: A new active learning strategy for crf-based named entity recognition](#). *Neural Processing Letters*, 54:2433–2454.
- Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. [Dr.ICL: Demonstration-retrieved in-context learning](#). In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*.
- Vijit Malik, Atul Pande, and Anirban Majumder. 2025. [Iclforge: Enhancing in-context learning with evolutionary algorithms under budgeted annotation](#). In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management, CIKM '25*, page 5931–5938, New York, NY, USA. Association for Computing Machinery.
- Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. [Active learning principles for in-context learning with large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5011–5034, Singapore. Association for Computational Linguistics.
- Costas Mavromatis, Balasubramaniam Srinivasan, Zhengyuan Shen, Jiani Zhang, Huzefa Rangwala, Christos Faloutsos, and George Karypis. 2023. Which examples to annotate for in-context learning? towards effective and efficient selection. *arXiv preprint arXiv:2310.20046*.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. [Can a suit of armor conduct electricity? a new dataset for open book question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.
- Neel Nanda, Senthooran Rajamanoharan, János Kramár, and Rohin Shah. 2023. Fact finding: Attempting to reverse-engineer factual recall on the neuron level. In *Alignment Forum*, volume 6.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. *Transformer Circuits Thread*. <https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html>.
- Chengwei Qin, Aston Zhang, Chen Chen, Anirudh Dagar, and Wenming Ye. 2024. [In-context learning with iterative demonstration selection](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7441–7455, Miami, Florida, USA. Association for Computational Linguistics.
- Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. [Learning to retrieve prompts for in-context learning](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.
- Stephanie Schoch and Yangfeng Ji. 2025. [Monte Carlo sampling for analyzing in-context examples](#). In *The Sixth Workshop on Insights from Negative Results in NLP*, pages 63–78, Albuquerque, New Mexico. Association for Computational Linguistics.
- Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.
- Lee Sharkey, Dan Braun, and Beren Millidge. 2022. [\[interim research report\] taking features out of superposition with sparse autoencoders](#).
- Artem Shelmanov, Vadim Liventsev, Danil Kireev, Nikita Khromov, Alexander Panchenko, Irina Fedulova, and Dmitry V. Dylov. 2019. [Active learning with deep pre-trained models for sequence tagging of clinical and biomedical texts](#). In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 482–489.
- Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2025. [Repetition improves language model embeddings](#). In *The Thirteenth International Conference on Learning Representations*.

- Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024. [Massive activations in large language models](#). In *First Conference on Language Modeling*.
- Pawel Swietojanski, Stefan Braun, Dogan Can, Thiago Fraga Da Silva, Arnab Ghoshal, Takaaki Hori, Roger Hsiao, Henry Mason, Erik McDermott, Honza Silovsky, Ruchir Travadi, and Xiaodan Zhuang. 2023. [Variable attention masking for configurable transformer transducer speech recognition](#). In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.
- Qwen Team. 2024. [Qwen2.5: A party of foundation models](#).
- Simon Tong. 2001. [Active learning: Theory and applications](#). Ph.D. thesis, Stanford University.
- Wilailuck Treerath and Fabrizio Pittorino. 2026. Active in-context learning for tabular foundation models. *arXiv preprint arXiv:2603.27385*.
- Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, and 16 others. 2020. [SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python](#). *Nature Methods*, 17:261–272.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. [Superglue: A stickier benchmark for general-purpose language understanding systems](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Xiaorong Wang, Ting Yang, Zhu Zhang, Shuo Wang, Zihan Zhou, Liner Yang, Zhiyuan Liu, and Maosong Sun. 2025. Monocle: Hybrid local-global in-context evaluation for long-text generation with uncertainty-based active learning. *arXiv preprint arXiv:2505.20195*.
- Xinyi Wu, Amir Ajorlou, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. 2024. [On the role of attention masks and layernorm in transformers](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 14774–14809. Curran Associates, Inc.
- Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. [Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1423–1436, Toronto, Canada. Association for Computational Linguistics.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.
- Bingsheng Yao, Guiming Chen, Ruishi Zou, Yuxuan Lu, Jiachen Li, Shao Zhang, Yisi Sang, Sijia Liu, James Hendler, and Dakuo Wang. 2024. [More samples or more prompts? exploring effective few-shot in-context learning for LLMs with in-context sampling](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1772–1790, Mexico City, Mexico. Association for Computational Linguistics.
- Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. [Compositional exemplars for in-context learning](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39818–39833. PMLR.
- Leihan Zhang and Le Zhang. 2020. [An ensemble deep active learning method for intent classification](#). In *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence, CSAI '19*, page 107–111, New York, NY, USA. Association for Computing Machinery.
- Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. [Active example selection for in-context learning](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

The Architecture Was Doing the Work: Why NLI Debiasing Results Do Not Generalise*

Prabhjot Singh

The University of Texas at Austin

prabhjot.singh@utexas.edu

Abstract

NLI debiasing research evaluates interventions on single architectures and datasets, leaving a critical assumption untested: do improvements reflect method quality or model capacity? We run 44 controlled experiments: eleven debiasing strategies across two architecturally distinct models (ELECTRA-small, RoBERTa-base) and two corpora (SNLI, MNLI), under identical conditions with Wilson 95% confidence intervals throughout. On ELECTRA-small, which achieves 87.4% in-domain accuracy, debiasing fails completely: all ten debiasing interventions fall within the baseline confidence interval on HANS (CoV=0.75%). On RoBERTa-base, methods show genuine sensitivity (CoV=10.76%, approximately $14\times$ higher), yet the median intervention trends toward degrading OOD accuracy on SNLI, short of conventional significance (Wilcoxon $p=0.065$), and method rankings across corpora are anti-correlated for ELECTRA (Spearman $\rho=-0.42$) but stable for RoBERTa ($\rho=0.75$). Contrastive augmentation swings 5.6 pp across corpora (+3.59 pp MNLI, -2.02 pp SNLI). The positive robustness-accuracy correlation on RoBERTa ($r=0.848$) inverts the assumed tradeoff: methods that improve HANS also preserve in-domain accuracy, while aggressive debiasers collapse both. The ELECTRA failure is not a pipeline artefact; the same codebase produces real gains on RoBERTa. It is a capacity failure: debiasing presupposes representational capacity that small architectures lack, a precondition the literature neither states nor tests.

1 Introduction

Every published NLI debiasing method carries an assumption nobody has tested: that a technique which works on one model and one dataset will work in general. We ran the experiment the field

has skipped and the answer is no. Applying the same eleven debiasing strategies, under identical conditions, to two architecturally distinct models, we find that method effectiveness is determined almost entirely by which architecture receives it, not by the method itself. The same code, seed, and data produce real improvements on one model and complete statistical silence on the other.

This is possible because the NLI debiasing literature has an evaluation blindspot it has not noticed. Methods are validated on single architectures, typically RoBERTa-large or BERT-base, on a single training corpus, without confidence intervals, paired significance tests, or cross-architecture validation. The community accumulates apparent progress while measuring a conflation of method effect and architecture effect that no single-architecture study can disentangle.

McCoy et al. (2019) demonstrated that NLI models exploit lexical heuristics rather than learning genuine inference; subsequent work proposed ensemble debiasing (He et al., 2019), Product-of-Experts (Clark et al., 2019), Debaised Focal Loss (Karimi Mahabadi et al., 2020), and data augmentation (Nie et al., 2020) as correctives. Every one was validated predominantly on one architecture. None asked whether the correction requires a minimum level of model capacity to function.

We ask that question across 44 controlled experiments: eleven debiasing strategies applied to ELECTRA-small (Clark et al., 2020) and RoBERTa-base (Liu et al., 2019) on both SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018), evaluated on HANS (McCoy et al., 2019). Every run shares an identical seed, optimiser, and evaluation protocol; every HANS accuracy carries a Wilson 95% CI; every cross-method comparison uses McNemar tests with Bonferroni correction. This is the first study to isolate the architecture variable across this many methods simultaneously.

On ELECTRA-small (87.4% in-domain on

*Code at: <https://github.com/prabhjotschugh/nli-debiasing-generalisation-study>

SNLI, not a trivially weak baseline), every intervention fails: CoV=0.75%, all ten methods within the baseline CI (Wilcoxon $p = 0.070$ SNLI, $p = 0.557$ MNLI). On RoBERTa-base, methods move but inconsistently: the median trends negative though not conventionally significant, and DFL ($\gamma = 3.0$) collapses in-domain accuracy by 22.96 pp while destroying robustness. Rankings reverse across corpora for ELECTRA but are stable for RoBERTa, underscoring that even RoBERTa’s genuine method sensitivity does not translate into architecture-independent rankings. The robustness-accuracy correlation on RoBERTa is strongly positive ($r = 0.848$), inverting the assumed tradeoff.

Our contributions: (1) the first controlled cross-architecture, cross-corpus evaluation of eleven NLI debiasing methods with full statistical reporting; (2) a novel failure mode where bias-model-dependent methods silently collapse to plain cross-entropy under insufficient model capacity; (3) dataset–method interactions (5.6 pp reversal) invisible to single-corpus evaluation; and (4) a positive robustness–accuracy correlation ($r = 0.848$) inverting the assumed tradeoff. These findings motivate four evaluation standards we argue should be non-negotiable for future debiasing work: cross-architecture validation on at least two structurally distinct models, Wilson confidence intervals with paired significance tests, model capacity reported as a moderating variable, and HANS per-heuristic breakdowns rather than a single aggregate score. We return to these in §4.

2 Experimental Setup

Models. ELECTRA-small

(google/electra-small-discriminator; 14M parameters) is pretrained via Replaced-Token-Detection, optimising for local token-level prediction.

RoBERTa-base (FacebookAI/roberta-base; 125M parameters) is pretrained via Masked Language Modelling, encouraging broader contextual representations. Both are fine-tuned as 3-class sequence classifiers.

Datasets. Training: SNLI (Bowman et al., 2015) (549K) and MNLI (Williams et al., 2018) (393K); unresolved labels filtered. In-domain evaluation uses SNLI test and MNLI validation-matched. OOD robustness uses HANS (McCoy et al., 2019): 30K binary examples across lexical overlap, subsequence, and constituent heuristics (10K each); per-heuristic breakdowns reported throughout.

Methods. 11 strategies share a unified codebase to eliminate confounders, grouped into 4 families:

- *Degenerate reference* - **Hypothesis-Only**: trained on hypothesis text alone; establishes the artifact ceiling.
- *Loss modification* - **Label Smoothing** ($\epsilon = 0.15$); **Confidence Penalty** ($\beta = 0.1$) (Pereyra et al., 2017) via $L = \text{CE} - \beta \cdot H(p)$, first applied here to NLI debiasing.
- *Bias-model-based* - **Ensemble Debiasing** ($\alpha = 0.5$) (He et al., 2019); **Product-of-Experts** ($\alpha = 0.5$) (Clark et al., 2019); **DFL** (Karimi Mahabadi et al., 2020) at aggressive ($\gamma = 3.0$, 1 ep) and conservative ($\gamma = 2.0$, 2 ep) settings, epoch counts fixed a priori to bound aggressive down-weighting rather than tuned per run. All use Hypothesis-Only as the bias model.
- *Data augmentation* - **ANLI Augmentation** (25% adversarial (Nie et al., 2020)); **Contrastive Augmentation** (150K rule-based contradictions via negation, antonym, quantifier, number swap); **Aggressive Augmentation** (same pipeline at 30% train size).

Training and statistics. All runs: AdamW, $\text{lr} = 2\text{e-}5$, weight decay 0.01, linear warmup (10%), seed 42 (Python/NumPy/PyTorch/CUDA), BF16 where supported. Default 3 epochs; DFL uses 1–2 epochs; contrastive augmentation uses 5 epochs. Batch sizes 32–256 per GPU memory; dual-input methods use halved batches. All HANS accuracies carry Wilson 95% CIs; Wilcoxon signed-rank tests assess systematic improvement (ten methods vs. baseline per model $\times$ corpus); Spearman ρ measures ranking consistency; McNemar tests with Bonferroni correction for pairwise significance. The *premise-irrelevance gap* (full-model minus Hypothesis-Only HANS accuracy) diagnoses premise conditioning capacity.

3 Results

3.1 Baseline Performance and Premise-Irrelevance Gap

Table 1 reports all 44 results. Both models are fully functional in-distribution: ELECTRA achieves 87.41% on SNLI ([86.74, 88.05]) and 79.37% on MNLI ([78.56, 80.16]); RoBERTa achieves 91.50% ([90.93, 92.04]) and 87.96% ([87.30, 88.59]). On HANS, ELECTRA collapses to near-random: 49.84% on SNLI and 51.87%

Method	ELECTRA-small				RoBERTa-base			
	SNLI		MNLI		SNLI		MNLI	
	In-Dom.	HANS Δ	In-Dom.	HANS Δ	In-Dom.	HANS Δ	In-Dom.	HANS Δ
Baseline (P+H)	87.41	49.84	79.37	51.87	91.50	67.21	87.96	70.25
Hypothesis-Only	70.86	51.10 +1.25	61.15	51.54 -0.33	70.86	50.13 -17.08	61.15	51.73 -18.53
<i>Loss Modification</i>								
Label Smoothing	87.39	49.83 -0.01	79.35	51.88 +0.01	91.64	65.27 -1.94	87.90	69.96 -0.29
Confidence Penalty	87.41	49.84 +0.00	79.37	51.90 +0.03	91.50	66.53 -0.68	87.78	70.65 +0.40
<i>Bias-Model-Based</i>								
Ensemble Debiasing	87.38	50.11 +0.27	79.34	51.51 -0.36	91.25	63.93 -3.28	87.36	67.92 -2.33
Product-of-Experts	87.40	49.66 -0.18	79.36	52.25 +0.38	90.89	65.98 -1.23	87.26	72.12 +1.86
DFL $\gamma=2.0$	87.39	49.98 +0.14	79.35	51.43 -0.44	76.89	61.99 -5.22	81.92	67.63 -2.62
DFL $\gamma=3.0$	87.38	50.11 +0.27	79.33	51.58 -0.29	68.54	51.58 -15.63	75.17	61.72 -8.53
<i>Data Augmentation</i>								
ANLI Augmentation	87.42	49.99 +0.15	79.38	52.11 +0.24	91.52	73.06 +5.85	87.87	73.43 +3.18
Contrastive Aug.	87.40	49.95 +0.10	79.36	52.31 +0.44	91.05	65.19 -2.02	87.58	73.84 +3.59
Aggressive Aug.	87.39	49.90 +0.06	79.34	50.91 -0.96	91.39	67.66 +0.45	87.64	71.62 +1.37

Table 1: In-domain and HANS accuracy (%) with Δ vs. baseline for all 44 settings.

on MNLI, with entailment bias of 99.56% and 94.83% respectively. The premise-irrelevance gap (full model minus Hypothesis-Only) is diagnostic: ELECTRA’s full model performs no better than a model discarding the premise entirely (-1.26 pp SNLI, $+0.33$ pp MNLI), while RoBERTa’s gap is $+17.08$ pp and $+18.52$ pp. ELECTRA learns nothing from the premise; RoBERTa genuinely conditions on it. Hypothesis-Only in-domain scores are identical across architectures as both exploit the same hypothesis-text artifacts.

3.2 ELECTRA-small: Complete Debiasing Failure

On ELECTRA/SNLI, HANS accuracy across all ten debiasing methods spans 49.66%–51.10%, entirely within the baseline CI of [49.28, 50.41]; on ELECTRA/MNLI, all ten fall within [51.30, 52.44]. The coefficient of variation is 0.75% (SNLI) and 0.79% (MNLI): the entire method distribution is tighter than a single 95% CI. Wilcoxon tests confirm no systematic improvement: SNLI $W = 7.0$, $p = 0.070$; MNLI $W = 21.0$, $p = 0.557$. The sole exception (Hypothesis-Only, 51.10%) discards the premise entirely; its ranking first is itself the finding. Aggressive Augmentation on MNLI (50.91%) falls *below* the CI lower bound: the one method that moves ELECTRA does so negatively.

3.3 RoBERTa-base: Real but Inconsistent Effects

RoBERTa shows genuine method sensitivity: CoV is 10.76% on SNLI and 9.42% on MNLI, roughly

$12\text{--}14\times$ higher than ELECTRA’s 0.75% and 0.79%. On SNLI, seven of ten methods hurt HANS accuracy; median delta -1.98 pp, a trend that falls short of conventional significance (Wilcoxon $W = 9.0$, $p = 0.065$). ANLI augmentation is the standout: $+5.85$ pp on SNLI (73.06%) and $+3.18$ pp on MNLI (73.43%), with the largest constituent gains ($+9.26$ pp SNLI, $+5.09$ pp MNLI), preserving in-domain accuracy exactly (91.52% vs. 91.50% baseline). DFL ($\gamma = 3.0$) is the sharpest failure: HANS collapses to 51.58% (-15.63 pp) while in-domain falls from 91.50% to 68.54% (-22.96 pp). Contrastive augmentation gains $+3.59$ pp on MNLI but loses -2.02 pp on SNLI, a 5.6 pp reversal with no change in model, method, or hyperparameters. Spearman $\rho = -0.42$ ($p = 0.20$) for ELECTRA and $\rho = 0.75$ ($p = 0.009$) for RoBERTa: ELECTRA rankings are anti-correlated noise across corpora; RoBERTa rankings are stable and real.

3.4 Robustness-Accuracy Correlation

The $\Delta\text{HANS}\text{--}\Delta\text{in-domain}$ correlation is $r = -0.096$ for ELECTRA, consistent with its flat method distribution, and $r = 0.848$ for RoBERTa. On RoBERTa, the assumed tradeoff inverts: methods that improve HANS preserve in-domain accuracy; methods that aggressively suppress bias (DFL, Ensemble Debiasing) lose on both axes. ANLI augmentation ($+5.85$ pp HANS, zero in-domain cost) and DFL $\gamma = 3.0$ (-15.63 pp HANS, -22.96 pp in-domain) define the poles.

3.5 Per-Heuristic Breakdown (RoBERTa-base)

Table 2 shows HANS per-heuristic accuracy for RoBERTa-base across both corpora. ELECTRA per-heuristic results are omitted: all values cluster within 49–56% across all methods and heuristics, consistent with the flat CoV reported in §3. The breakdown reveals that aggregate HANS accuracy obscures qualitatively distinct failure modes.

Method	SNLI			MNLI		
	Lex	Sub	Con	Lex	Sub	Con
Baseline	82.92	61.18	57.53	81.95	64.00	64.81
Label Smooth.	82.13	59.10	54.58	81.53	63.91	64.45
Conf. Penalty	82.63	60.39	56.58	81.87	64.59	65.49
Ensemble	76.74	58.30	56.75	78.64	61.75	63.38
PoE	85.55	57.52	54.87	85.18	65.17	66.00
DFL $\gamma=2.0$	75.86	56.31	53.79	75.46	61.69	65.74
DFL $\gamma=3.0$	51.98	50.94	51.83	63.29	60.43	61.45
ANLI Aug.	87.88	64.52	66.79	84.14	66.26	69.90
Contrastive	79.41	57.38	58.79	86.84	65.34	69.34
Aggressive	85.27	60.26	57.45	81.37	64.18	69.32
Hyp-Only	49.88	50.27	50.23	50.08	55.09	50.01

Table 2: HANS per-heuristic accuracy (%) for RoBERTa-base. Lex = lexical overlap, Sub = subsequence, Con = constituent.

4 Analysis and Discussion

The capacity hypothesis. Bias-model-based methods require a functional feedback loop: the bias model flags artifact-reliant examples so the main model can down-weight them. On ELECTRA, this loop fails at the source. The hypothesis-only bias model achieves $\sim 51\%$ on HANS, near chance, so the debiasing loss reduces to plain cross-entropy; the machinery runs but steers with noise. Bias-free methods fail too: label smoothing and confidence penalty cannot inject cross-sentence reasoning capacity the architecture lacks; augmentation cannot force premise-sensitivity into a model that consistently ignores the premise. The premise-irrelevance gap (-1.26 pp on SNLI) makes this structural: the capacity ceiling is set before any intervention begins.

Pretraining as a confound. The gap is not solely parameter count. ELECTRA’s Replaced-Token-Detection optimises for local token-level prediction; RoBERTa’s Masked Language Modelling requires broader cross-sentence representations. This inductive bias difference likely restricts ELECTRA’s ability to build premise-hypothesis repre-

sentations independent of size. A method validated exclusively on MLM-pretrained encoders cannot be assumed to transfer to RTD-pretrained models, an unstated precondition of almost every published debiasing result.

The publication selection effect. The approximately $14\times$ CoV difference (0.75% vs. 10.76%) exposes the problem precisely. A researcher running only RoBERTa observes real sensitivity and publishes a working method. A researcher running only ELECTRA observes flat noise and has nothing to publish. The published record consists of the first kind of result. Same codebase, same seed, same data: one architecture moves, the other does not. What the field calls method effectiveness is the interaction of method and architecture, inseparable without cross-architecture controls.

Dataset-method interactions. The 5.6 pp contrastive augmentation reversal across SNLI and MNLI reflects a systematic interaction between generated examples and corpus artifact distributions. SNLI’s densely patterned crowdsourced hypotheses may cause rule-based contradictions to reinforce existing artifacts rather than disrupt them; MNLI’s multi-genre hypotheses present a more varied artifact landscape the same contradictions successfully target. ANLI augmentation’s constituent gains ($+9.26$ pp SNLI, $+5.09$ pp MNLI) follow the same logic: adversarially collected examples target the failure modes most prevalent in the corpus they were built against. These interactions are invisible to single-corpus evaluation.

The positive robustness-accuracy correlation. ANLI augmentation introduces constructive gradient signal where heuristics misfire, expanding the decision boundary without disrupting in-distribution learning. DFL aggressively down-weights artifact-aligned examples; when the model cannot find alternative generalisable features, it destroys both axes simultaneously. Within the scope of the artifacts HANS captures, the robustness-accuracy tradeoff is not axiomatic; it is a property of overly aggressive methods, not of debiasing in general. We do not claim this extends to bias patterns HANS does not test.

5 Conclusion

We set out to test an assumption the NLI debiasing literature has never examined: that a method

validated on one architecture and one corpus generalises beyond it. Across 44 controlled experiments it does not. Debiasing method effectiveness is not a property of the method in isolation; it is jointly determined by the method and the architecture receiving it, and the two cannot be separated without exactly the cross-architecture design the field has been skipping.

The practical consequence is a publication selection effect: a researcher who happens to run RoBERTa observes real, if inconsistent, method sensitivity and has a paper; a researcher who happens to run ELECTRA observes flat noise and has nothing to publish, even on an identical codebase with an identical seed. What the field records as method effectiveness is this selection artefact as much as it is genuine progress. The clearest evidence against a simple robustness-accuracy trade-off is ANLI augmentation, which improves HANS accuracy at essentially zero in-domain cost, set against DFL at its aggressive setting, which damages both simultaneously; the tradeoff, where it appears at all, is a symptom of method aggressiveness rather than an inherent property of debiasing.

We do not read these results as a reason to abandon debiasing research, but as a case for conducting it with the rigour its evaluation infrastructure has so far lacked: cross-architecture validation on at least two structurally distinct models, Wilson confidence intervals with paired significance tests, model capacity reported as a moderating variable, and HANS per-heuristic breakdowns rather than a single aggregate number. Without these controls, a positive result is not wrong so much as impossible to interpret.

Limitations.

ELECTRA-small and RoBERTa-base differ in both parameter count (14M vs. 125M) and pretraining objective (Replaced-Token-Detection vs. Masked Language Modelling), so the two factors are confounded in our design: we cannot conclusively attribute ELECTRA’s failure to representational capacity alone rather than to properties of RTD pretraining specifically. We favour the capacity explanation because it is consistent with the premise-irrelevance gap, a within-model diagnostic that does not depend on comparing across architectures, but isolating size from pretraining objective would require a small MLM-pretrained model (e.g., BERT-tiny) run under the same protocol, which we

leave to future work. Results may not generalise to larger architectures (RoBERTa-large, DeBERTa) or non-English NLI. All runs use a single random seed; multiple seeds would further strengthen stability claims. The contrastive pipeline is rule-based, and HANS covers three heuristics but does not exhaust NLI artifact space.

References

- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. [Don’t take the easy way out: Ensemble based methods for avoiding known dataset biases](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [Electra: Pre-training text encoders as discriminators rather than generators](#). *Preprint*, arXiv:2003.10555.
- He He, Sheng Zha, and Haohan Wang. 2019. [Unlearn dataset bias in natural language inference by fitting the residual](#). In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.
- Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. [End-to-end bias mitigation by modelling biases in corpora](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.
- R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. [Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.
- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. [Adversarial NLI: A new benchmark for natural language understanding](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.
- Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *ICLR Workshop*. ArXiv:1701.06548.
- Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

When Learned Context Planning Fails to Beat Strong Retrieval: A Controlled Study of Planning, Routing, and Reranking for Long-Context QA

Yingrui Li¹ Han Chen²

¹Independent Researcher, United States ²Independent Researcher, United States

emmalyr0404@gmail.com, hc414t@gmail.com

Abstract

Learned context planning selects evidence atoms before an answer model reasons over them. We test whether this learned selection improves long-context multiple-choice QA after strong retrieval, routing, budgeted-selector, and reranking controls. Our primary diagnostic uses all 503 LongBench-v2 MCQ questions with Qwen2.5-7B-Instruct. The planner is SFT-trained on outcome-selected traces from 140 training and 28 development questions; because the 503-question analysis includes those questions, it is partly transductive. At an 18k-character budget, anchored hybrid retrieval reaches 36.18% accuracy and BM25 reaches 35.98%, while the best direct planner-guided method reaches 34.19%. On the untouched 152-question test split, anchored hybrid remains higher (42.11% versus 36.84%). Leakage-safe routers cannot convert a large oracle gap. Under tight budgets, the best planner is ahead by only 0.40 points at 6k and loses at 9k; planner-guided reranking has a +1.79-point estimate at 6k with a paired interval crossing zero and ties the control at 9k. Packing-order and score-flatness analyses did not identify a stable mechanism. Under this setup, learned planning is a weak relevance signal rather than a replacement for strong retrieval.

1 Introduction

Long-context question answering systems must decide which parts of a large input to show to an answer model. Retrieval-augmented generation (RAG) usually ranks chunks with lexical, dense, or hybrid retrieval, then packs the top results into context (Robertson and Zaragoza, 2009; Lewis et al., 2020; Reimers and Gurevych, 2019). A learned context planner offers a different interface: retrieve a candidate set, ask a trained model to select evidence atoms, and answer from the selected or

planner-prioritized context. Here, planning means evidence selection; the planner outputs atom IDs rather than multi-step reasoning traces.

This idea is plausible, but a weak comparison can make it look stronger than it is. A planner may beat prompted planning or a direct baseline while still losing to BM25, and a large oracle routing bound may not imply a deployable inference-time router.

We ask: *when does learned evidence planning survive strong retrieval and reranking controls?* We answer this question on LongBench-v2 multiple-choice QA (Bai et al., 2024). Learned planning changes the evidence set, helps some examples, and produces a small 6k-character rerank signal. However, in this setting it does not robustly beat strong retrieval, routing, budgeted selection, or reranking controls.

We show that (i) direct planner-guided construction loses to BM25 and anchored hybrid retrieval, (ii) oracle complementarity does not become a leakage-safe router, (iii) low-budget planner gains shrink under stronger selectors, and (iv) the remaining planner-rerank signal is statistically unsupported. We additionally audit train–evaluation separation, held-out behavior, domain variation, packing order, and retrieval-score mechanisms. The practical takeaway is that planner gains should be checked against strong retrieval, routing, budgeted selection, and reranking controls before being attributed to planning.

2 Experimental Setup

Task, split, and executor. We deterministically split the 503 LongBench-v2 MCQ questions into 301 training, 50 development, and 152 test questions (split seed 20260225). Each example has a long context, question, choices, and a gold let-

ter; a run is correct when the parsed output letter matches the gold. The answer model is Qwen2.5-7B-Instruct (Yang et al., 2024). Our primary diagnostic reports all 503 questions and therefore includes the 168 questions used to construct planner SFT train/development records. It is partly transductive and is not a conventional held-out planner-generalization estimate. We separately report the untouched 152-question test partition.

Atoms and retrieval controls. An *evidence atom* is a deterministic character-bounded chunk. Multi-document contexts are split on document separators or headers, then chunked into 2,000-character atoms with 200-character overlap; other contexts use 8,000-character atoms with 1,000-character overlap. A method ranks or selects atoms and packs their full text under an 18k, 9k, or 6k character budget. BM25 uses an alphanumeric tokenizer. Dense retrieval uses sentence-transformers/all-MiniLM-L6-v2; hybrid retrieval applies reciprocal-rank fusion. *Anchored hybrid* places top BM25 atoms before hybrid fill. Budgeted controls include no-anchor and weighted hybrid packing and MMR. Non-planner rerankers score the same BM25+dense candidate pool with cross-encoder/ms-marco-MiniLM-L-6-v2 (Wang et al., 2020).

Trace mining and planner SFT. For each training or development question, the base Qwen model samples six plans from a 30-atom retrieval candidate pool (temperature 0.7, top- p 0.9, at most eight selected atoms, and an 18k-character execution budget). We deduplicate plans by selected atom set, execute each unique plan with the same base answer model, and assign binary reward from answer correctness. For SFT, we retain one reward-one plan per question, preferring fewer atoms, shorter context, and earlier sample order. This yields 140 SFT training rows and 28 development rows, with zero test-ID overlap.¹ We train a Qwen2.5-7B-Instruct LoRA adapter (Hu et al., 2022) for three epochs with rank 16, alpha 32, dropout 0.05, learning rate 5×10^{-5} , per-device batch size 2, gradient accumulation 8, bfloat16 precision, and seed 111, using Transformers and PEFT (Wolf et al., 2020;

Mangrulkar et al., 2022). The final adapter is the global-step-24 state; no development checkpoint selection occurs because training ends before the configured evaluation interval.

At inference, the planner receives only the question, choices, and retrieved candidate atoms. *Planner-guided hybrid* packs planner-selected atoms before BM25 anchors and hybrid fill; *planner-rerank* packs planner-selected atoms before cross-encoder reranked fill; *Planner+BM25* uses BM25 fill. We also report an earlier-adapter sensitivity check.

Routers, statistics, and artifacts. An oracle router chooses post hoc using gold correctness and measures complementarity rather than deployable performance. Leakage-safe routers choose among context-construction policies using only inference-available features under stratified 5-fold and leave-domain-out validation. We report accuracy, paired wins/losses, exact two-sided sign-test p -values, and 95% intervals. The public repository provides evaluation and analysis code, portable configurations, split IDs, environment records, checksums, text-free per-example predictions, atom IDs, retrieval/reranker ranks and scores, planner-selected and packed IDs, and paired-test outputs.² Released scripts regenerate the main, split, domain, packing-order, paired-sign, and compact mechanism analyses; router and selector results and some intervals are provided as audited summaries. Raw LongBench-v2 contexts, base-model weights, and adapter weights are not redistributed. We do not claim bitwise reproduction of historical GPU inference, and latency is descriptive only.

3 Results

3.1 Direct Planning Loses to Retrieval

Table 1 gives the full-budget result. Anchored hybrid retrieval is best at 36.18%, and BM25 is nearly tied at 35.98%. The best direct planner-guided method reaches 34.19%. Planner+BM25 is lower than BM25 by 3.78 points (56 BM25-only wins versus 37 planner-only wins; $p = 0.061$). Thus, direct planner-guided methods do not beat strong retrieval in this setup.

Held-out robustness. On the 152 questions with no SFT train/development overlap, anchored hybrid reaches 42.11% and planner-guided anchored hybrid reaches 36.84% (difference -5.26 points;

¹The submission version reported 262 retained traces in a merged SFT archive. A release audit could not reproduce that count from the audited artifacts of the final adapter; the verified counts are the 140 training and 28 development rows (168 unique questions) reported here.

²<https://github.com/aapplepku/context-planning-vs-retrieval>

Method	Acc.	95% CI	Pair p
Hybrid anchor	36.18	[32.10, 40.47]	—
BM25	35.98	[31.91, 40.27]	1.000 ^a
Planner+hybrid anchor	34.19	[30.18, 38.45]	0.399 ^b
Planner+BM25	32.21	[28.27, 36.41]	0.061 ^c
Dense	30.22	[26.37, 34.37]	—

Table 1: Full 18k-context results on 503 questions. CIs are Wilson intervals. ^aBM25 vs. hybrid anchor. ^bPlanner+hybrid anchor vs. hybrid anchor. ^cPlanner+BM25 vs. BM25.

Domain (n)	BM25	Anchor	P+anchor	Δ
Code repo. (50)	40.00	40.00	34.00	-6.00
Long ICL (81)	34.57	34.57	35.80	+1.23
Structured (33)	39.39	39.39	36.36	-3.03
Dialogue hist. (39)	25.64	25.64	25.64	0.00
Multi-doc QA (125)	30.40	31.20	30.40	-0.80
Single-doc QA (175)	41.14	41.14	37.71	-3.43

Table 2: Full-budget accuracy (%) by LongBench-v2 domain. Δ is planner-guided anchored hybrid minus anchored hybrid. These subgroup results are descriptive.

12 planner wins, 20 losses, 120 ties; $p = 0.215$; paired-bootstrap 95% CI $[-12.50, 1.97]$ points). This lower-powered subset does not provide evidence that planning robustly exceeds the retrieval control.

At full budget, planner-guided anchored hybrid exceeds its non-planner counterpart in one of six domains, ties in one, and trails in four (Table 2). Several domain subsets are small, so we do not treat these differences as confirmatory.

3.2 Oracle Routing Does Not Become a Router

Oracle routing shows complementary errors: an oracle over hybrid anchor and planner+hybrid reaches 46.52%, and an all-method oracle reaches 56.26%. These are post-hoc upper bounds, not deployable policies. Table 3 separates them from leakage-safe routers using task/domain meta-data, context-length statistics, retrieval-score summaries, planner parse/selection metadata, and choice-confidence features. Across stratified 5-fold and leave-domain-out validation, logistic regression, shallow random forests, histogram gradient boosting, domain-only rules, and default-train base-lines fail to improve over the best single method.

3.3 Budget and Rerank Controls Erase Most Gains

A weaker 6k comparison initially favored planner-guided hybrid over anchored hybrid by 1.99 points.

Policy	Acc.	Gain	Prot.
Best single: hybrid anchor	36.18	0.00	fixed
Oracle: hybrid/planner	46.52	+10.34	post hoc
Oracle: all methods	56.26	+20.08	post hoc
Pair routers	36.18	0.00	5-fold/LDO
BM25-vs-planner routers	35.98	-0.20	5-fold/LDO
Five-way routers	35.98	-0.20	5-fold/LDO
Domain router	36.38	+0.20	same-data

Table 3: Oracle bounds and routers. Fixed = best single method; post hoc = oracle using gold correctness; 5-fold/LDO = stratified 5-fold and leave-domain-out validation; same-data = optimistic in-sample rule. The domain router selects, per domain, the better of hybrid anchor and planner-guided anchored hybrid using in-sample accuracy; its +0.20-point gain corresponds to one additional question.

Set	Bgt	Ctrl	Plan	Δ [95% CI]
Sel.	6k	32.80	33.20	+0.40 [-4.01,+4.81]
Sel.	9k	35.59	34.39	-1.19 [-5.43,+3.04]
Rerank	6k	31.41	33.20	+1.79 [-2.58,+6.16]
Rerank	9k	34.19	34.19	0.00 [-4.37,+4.17]

Table 4: Best planner method versus best non-planner control at each budget. Paired p -values are 0.930, 0.645, 0.478, and 1.000 in row order. Values are percentage points.

We therefore add stronger budgeted selectors and reranker controls. Table 4 reports the decisive comparisons. Against the best same-budget non-learned selector, the 6k planner gain is only +0.40 points with a wide interval and $p = 0.930$; at 9k the planner loses by 1.19 points. Against strong non-planner rerank controls, planner-rerank has a +1.79-point estimate at 6k, but the paired interval crosses zero and $p = 0.478$. At 9k, it ties the best non-planner reranker.

The earlier sensitivity adapter raises the 6k planner-rerank score to 34.79%, but still lacks paired support ($p = 0.146$; CI $[-0.99, +7.75]$ points) and does not beat the 9k control. Its selected-atom Jaccard with the final adapter is 0.881, indicating similar selections but not a robust accuracy gain.

3.4 Selection, Order, and Score Mechanisms

Holding the planner-selected set fixed, hybrid-retrieval order changes accuracy relative to planner order by -1.19 points at 6k ($p = 0.461$) and

+0.99 points at 9k ($p = 0.511$). The reversal does not support a systematic packing-order advantage. We also compare planner-selected IDs with BM25 and cross-encoder rankings, measure high-ranked-atom displacement, and test four predeclared score-flatness statistics. Planner-reranker Jaccard@10 is 0.084 for planner wins versus 0.097 for losses at 6k, and 0.082 versus 0.111 at 9k; planner contexts exclude an average 8.5–9.3 of the reranker’s top-10 atoms. None of 92 win–loss tests remains significant after Benjamini–Hochberg correction. Thus, the outputs do not support a stable claim that planner wins occur specifically when retrieval scores are flatter. These are rank and context-membership analyses, not evidence-recall measurements, because LongBench-v2 provides no gold atom-level evidence labels for this study.

4 Discussion and Conclusion

When compared only with prompted or heuristic planning and direct context baselines, a learned planner can look useful. After adding BM25, anchored hybrid packing, and reranking controls, most apparent gains disappear, even though the planner remains trainable and sometimes changes the evidence set helpfully. Paired analysis is essential: the +1.79-point 6k estimate comes from 68 planner wins and 59 losses among 503 questions. Under the observed discordance, confirming a +1.5-point paired effect would require roughly 8,800 examples and a +2.0-point effect roughly 4,900 examples.

In this bounded LongBench-v2 MCQ study, learned context planning is better viewed as a weak relevance feature than as a replacement for retrieval. Future studies should compare against anchored hybrid retrieval, budgeted selectors, leakage-safe routing, and strong rerankers before interpreting gains as evidence of planning ability.

Limitations

Our conclusions are restricted to LongBench-v2 MCQ, Qwen2.5-7B-Instruct, one outcome-supervised planner recipe, one overlapping character-window atomization scheme, the evaluated lexical/dense retrievers, and the MiniLM cross-encoder. The all-503 diagnostic is partly transductive for the 168 questions represented in SFT train/development data; the disjoint test analysis has only 152 questions and limited power for small paired effects. We do not evaluate open-ended QA,

other model scales, alternative supervision sources, or real-world deployment, and we have no external replication. A larger or differently mined trace set could change the outcome. The historical run pre-dates strict locked-run enforcement and immutable model-revision pinning, and latency is descriptive rather than a portable efficiency claim. LongBench-v2 supplies no gold atom-level evidence labels for this analysis, so mechanism results use atom IDs, ranks, scores, and displacement rather than evidence recall. These results do not imply that planners are generally ineffective; they show that this planner did not robustly beat strong retrieval controls under the studied conditions.

Acknowledgments

We thank the Insights 2026 reviewers for their careful comments and suggestions. Generative AI assistants were used to help draft and revise portions of this manuscript and to help prepare the analysis and artifact-release scripts. All experimental results, statistics, and claims were verified by the authors against the released artifacts, and the authors take full responsibility for the content.

References

- Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*.
- Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. <https://github.com/huggingface/peft>.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*, pages 3982–3992. Association for Computational Linguistics.

- Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, and others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP: System Demonstrations*, pages 38–45.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

No Optimal Language Set Exists for Multilingual Instruction Tuning: Insights from a Linguistically-Informed Study

Gürkan Soykan^{*1}, Gözde Gül Şahin^{2,3,4}

¹ Information Technology Group, Wageningen University & Research, Netherlands

² Friedrich-Alexander-Universität Erlangen-Nürnberg, Intelligent Language Systems

³ Computer Engineering Department, Koç University, Istanbul, Turkey

⁴ KUIS AI Lab, Istanbul, Turkey

<https://gglab-ku.github.io/>

Abstract

Multilingual instruction tuning (MIT) is challenged by the curse of multilinguality, data scarcity, and high computational cost. A natural hypothesis is that carefully selecting a linguistically diverse set of languages yields universally better models. We test this systematically by evaluating linguistically-informed selection strategies—based on typological, geographical, semantic, and learned features—against random baselines across three model families (mGPT, mT5-xl, BLOOM) and five multilingual benchmarks. Our key negative finding is that no universal language selection strategy emerges in our fixed-budget setting: performance is strongly task- and model-dependent, and adding more languages beyond a modest threshold triggers degradation consistent with the curse of multilinguality. We discuss implications for MIT data curation and the pitfalls of benchmark-averaged evaluation. All resources are publicly available at <https://github.com/GGLAB-KU/ling-informed-mit>.

1 Introduction

Multilingual language models exhibit uneven cross-lingual performance (Joshi et al., 2020), and instruction tuning with multilingual data has emerged as a practical remedy (Muennighoff et al., 2022). However, the curse of multilinguality (Conneau et al., 2019)—where per-language performance degrades as more languages compete for model capacity—complicates the pursuit of broad coverage, as do high computational costs and data scarcity.

A natural hypothesis follows: selecting the *right* linguistically diverse subset of languages should yield models that generalize well across languages and tasks. Prior work lends partial support—Chen et al. (2023a) and Shaham et al. (2024) show small

multilingual subsets can match or exceed monolingual tuning; Weber et al. (2024) confirm that parallel multilingual corpora improve cross-lingual instruction-following by up to 9.9% over monolingual training; and language features have aided cross-lingual transfer in related tasks (Lin et al., 2019; Üstün et al., 2020). Yet no prior work has systematically investigated whether linguistically-informed language *selection* yields a universally superior MIT strategy.

We fill this gap. Our central negative finding is that **no universal language selection strategy was observed in the fixed-budget MIT setting studied here**: the best choice depends on model family and evaluation task, which has important consequences for how the community evaluates and reports MIT results.

2 Experimental Setup

We use Bactrian-X (Li et al., 2023), a parallel multilingual instruction dataset with 3.4M instruction-response pairs across 52 languages. Following the budget-controlled setup of Chen et al. (2023a), we fix total training to 67k instances per experiment (4,786 per language for 14-language subsets). We select 14 languages per subset using k-means clustering ($k=14$) on different feature spaces: **GEO** (geographical vectors (Hammarström et al., 2023)), **TYPO** (typological vectors (Littell et al., 2017)), **SEM** (semantic typology embeddings (Chen et al., 2023b)), **LEARN** (learned language vectors (Malaviya et al., 2017)), **FAM** (one language per family), **RND** (random, baseline), and **ALL** (all 52 languages). Each is run with 3 random seeds. Table 1 lists the selected languages.

We instruction-tune three model families with LoRA (Hu et al., 2021): mGPT-1.3B (Shliazhko et al., 2024), mT5-XL-3.7B (Xue et al., 2021), and BLOOM-{1.7B, 3B, 7B} (Scao et al., 2022), eval-

^{*}E-mail: gurkan.soykan@wur.nl, Work is done during PhD studies at Koç University.

Table 1: Language subsets per strategy and seed (ISO codes).

Subset	Languages
FAM	az,en,fi,he,ja,ka,vi,ko,ml,mn,my,th,tl,xh ar,en,fi,id,ja,ka,ko,mn,sw,ta,th,tr,vi,zh af,az,et,he,ja,ka,ko,ml,mn,my,th,tl,vi,xh
GEO	af,bn,et,fr,gu,he,hr,id,ja,kk,mn,sw,ta,vi af,bn,et,fr,he,hr,id,ja,ka,kk,mn,ta,ur,vi af,bn,et,fr,he,hr,id,ja,kk,mn,sw,ta,ur,vi
LEARN	cs,hi,id,km,ko,lt,lv,my,nl,pt,sl,ta,tl,vi cs,fi,hr,km,ko,lt,lv,my,nl,pt,sl,ta,uk,vi cs,hi,id,ja,lt,lv,nl,pt,sl,sv,ta,tl,tr,vi
RND	cs,gu,hi,id,ko,lv,mk,ml,ps,pt,si,ta,vi,zh cs,en,et,fr,hi,hr,it,ja,ml,mn,mr,ne,pl,pt ar,az,de,en,es,et,he,hi,id,ml,ps,ru,sv,uk
SEM	gl,gu,ka,kk,ko,ml,pl,ps,si,sl,sv,tl,uk,vi en,fr,gl,gu,ka,kk,ko,pl,ps,sl,sv,tl,uk,vi en,gl,gu,ka,kk,ko,ml,pl,ps,sl,sv,tl,uk,vi
TYPO	az,bn,de,et,fa,hi,it,ja,mk,sw,ta,tl,vi,zh ar,az,bn,de,et,fa,it,ja,mk,sw,ta,th,ur,zh ar,az,bn,en,fi,he,hi,it,ja,mk,nl,sw,ta,th

uated zero-shot on XNLI (Conneau et al., 2018), PAWS-X (Yang et al., 2019), XCOPA (Ponti et al., 2020), XStoryCloze (Lin et al., 2021), and XWinograd (Tikhonov and Ryabinin, 2021) using the lm-evaluation-harness (Gao et al., 2023).

3 Results

3.1 No Single Strategy Wins Across All Tasks and Models

Table 2 presents our main results for BLOOM-7B, mGPT, and mT5-xl. The best-performing strategy varies substantially across models and tasks: GEO leads for BLOOM-7B on XStoryCloze while TYPO achieves the best average for BLOOM-7B overall, FAM is best on average for mT5-xl, and GEO or TYPO lead for mGPT on NLU tasks. Crucially, **no single subset consistently achieves the highest score across all model-task combinations**.

Improvements over the RND baseline are often not statistically significant. Paired t-tests (three seeds) show significant improvements only in specific model-task combinations—e.g., GEO vs. RND for mGPT on XNLI ($p \leq 0.05$) and XStoryCloze ($p \leq 0.05$), FAM vs. RND for mT5-xl on XCOPA ($p \leq 0.05$) and XWinograd ($p \leq 0.05$)—but these do not generalize across all benchmarks or model families. Meanwhile, several linguistically-informed subsets score *significantly lower* than RND in specific configurations (red

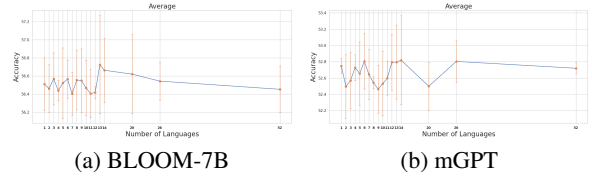


Figure 1: Average performance with confidence intervals when varying the number of GEO-selected languages. Performance peaks at 13–14 and declines beyond.

cells), underscoring that informed selection can also backfire. The effect sizes in Table 2 contextualise this: Cohen’s d_z expresses the mean gain over RND in units of its seed-to-seed variability (0.2/0.5/0.8=small/medium/large) and derives from the same paired t-tests. With $n=3$ seeds, $p \leq 0.05$ requires $|d_z| \geq 2.5$, so only very large effects can reach significance. Even these are model-specific (GEO: $d_z=3.6$ on mGPT vs. 0.5 on BLOOM-7B), and with three seeds individual d_z values are magnitude indicators rather than precise estimates.

For PAWS-X, no strategy produces significant improvement over RND in any model, and the base (non-tuned) model achieves the best PAWS-X score for both BLOOM-7B and mGPT—suggesting MIT may actively harm paraphrase identification performance.

3.2 More Languages Triggers Curse of Multilinguality

Figure 1 shows the effect of varying language count (1 to 52) using GEO on mGPT and BLOOM-7B. For BLOOM-7B, performance is stable up to 13–14 languages, peaks there, then declines consistently—replicating the curse of multilinguality (Conneau et al., 2019) under a fixed computational budget. The ALL setting (52 languages) never achieves best average performance in either model. mGPT shows wider confidence intervals but a similar trend: the 14-language default performs marginally better than larger configurations.

3.3 Cross-lingual Generalization to Unseen Languages is Model-Dependent

Figure 2 shows model performance on *unseen* languages (those in evaluation benchmarks but not in any training subset). While linguistically-informed subsets generally improve over the non-tuned base model for mGPT, mT5-xl, and BLOOM-3B, results diverge for larger BLOOM models. For BLOOM-

Table 2: Main results for BLOOM-7B, mGPT, and mT5-xl. **Bold**: highest mean per model-task. Color coding vs. RND baseline (paired t-test): **dark green** $p \leq 0.05$, **green** $0.05 < p \leq 0.1$, **light green** $0.1 < p \leq 0.15$ (significantly better); **dark red**, **red**, **light red** (significantly worse). “-” = base model (no instruction tuning). In the Avg. column, the parenthesised value is the paired effect size vs. RND (Cohen’s d_z).

Model	Subset	XNLI	XCOPA	XStoryCloze	XWinograd	PAWS-X	Avg. (d_z)
Bloom-7B	ALL	40.88 $\pm$ 0.25	58.18 $\pm$ 0.54	62.03 $\pm$ 0.13	73.71 $\pm$ 0.23	47.46 $\pm$ 0.82	56.45 $\pm$ 0.26 (-0.1)
	RND	41.63 $\pm$ 0.25	57.72 $\pm$ 0.86	61.73 $\pm$ 0.30	73.91 $\pm$ 0.31	47.36 $\pm$ 2.85	56.47 $\pm$ 0.69
	FAM	41.18 $\pm$ 0.67	57.95 $\pm$ 0.19	61.49 $\pm$ 0.72	74.00 $\pm$ 0.81	47.88 $\pm$ 2.29	56.50 $\pm$ 0.56 (0.1)
	GEO	41.24 $\pm$ 0.66	57.97 $\pm$ 0.39	62.15 $\pm$ 0.31	73.96 $\pm$ 1.69	47.99 $\pm$ 0.85	56.66 $\pm$ 0.35 (0.5)
	LEARN	41.26 $\pm$ 0.66	58.08 $\pm$ 0.22	61.96 $\pm$ 0.06	74.17 $\pm$ 0.29	46.88 $\pm$ 1.85	56.47 $\pm$ 0.21 (0.0)
	SEM	41.19 $\pm$ 0.54	58.03 $\pm$ 0.66	61.44 $\pm$ 0.69	74.16 $\pm$ 0.18	47.35 $\pm$ 0.47	56.43 $\pm$ 0.31 (-0.1)
	TYPO	41.63 $\pm$ 0.80	58.09 $\pm$ 0.54	61.94 $\pm$ 0.28	74.17 $\pm$ 1.08	47.75 $\pm$ 0.31	56.71 $\pm$ 0.24 (0.8)
mGPT	-	41.12	56.87	59.30	73.97	49.37	56.13
	ALL	41.23 $\pm$ 0.12	55.82 $\pm$ 0.14	55.98 $\pm$ 0.07	60.44 $\pm$ 0.30	50.14 $\pm$ 0.43	52.72 $\pm$ 0.07 (2.0)
	RND	40.71 $\pm$ 0.23	55.91 $\pm$ 0.62	55.97 $\pm$ 0.36	60.78 $\pm$ 0.20	48.71 $\pm$ 0.99	52.42 $\pm$ 0.36
	FAM	40.27 $\pm$ 0.08	55.82 $\pm$ 0.92	55.81 $\pm$ 0.85	60.88 $\pm$ 1.04	49.69 $\pm$ 1.72	52.49 $\pm$ 0.30 (0.7)
	GEO	41.01 $\pm$ 0.29	55.73 $\pm$ 0.56	56.35 $\pm$ 0.13	61.05 $\pm$ 0.44	49.97 $\pm$ 2.64	52.82 $\pm$ 0.55 (3.6)
	LEARN	40.64 $\pm$ 0.42	55.95 $\pm$ 0.47	56.15 $\pm$ 0.27	60.66 $\pm$ 1.03	48.92 $\pm$ 0.84	52.46 $\pm$ 0.38 (0.2)
	SEM	40.77 $\pm$ 0.21	55.61 $\pm$ 0.42	56.14 $\pm$ 0.34	60.57 $\pm$ 0.47	49.17 $\pm$ 2.17	52.45 $\pm$ 0.44 (0.4)
mT5-xl	TYPO	40.84 $\pm$ 0.31	55.93 $\pm$ 0.57	56.24 $\pm$ 0.44	60.39 $\pm$ 0.83	49.17 $\pm$ 2.06	52.51 $\pm$ 0.64 (0.7)
	-	40.90	55.04	54.43	60.55	50.30	52.24
	ALL	36.49 $\pm$ 0.69	53.64 $\pm$ 1.01	52.31 $\pm$ 0.21	50.39 $\pm$ 0.81	50.76 $\pm$ 1.15	48.72 $\pm$ 0.31 (-0.3)
	RND	36.86 $\pm$ 0.29	53.10 $\pm$ 0.48	52.46 $\pm$ 0.52	49.88 $\pm$ 0.75	51.74 $\pm$ 1.74	48.81 $\pm$ 0.55
	FAM	36.12 $\pm$ 0.44	53.72 $\pm$ 0.34	52.59 $\pm$ 0.34	51.01 $\pm$ 1.69	52.28 $\pm$ 0.76	49.14 $\pm$ 0.19 (1.4)
	GEO	36.91 $\pm$ 0.36	53.23 $\pm$ 0.23	52.61 $\pm$ 1.03	50.20 $\pm$ 0.76	52.11 $\pm$ 0.91	49.01 $\pm$ 0.50 (0.8)
	LEARN	36.69 $\pm$ 0.55	53.48 $\pm$ 0.53	52.62 $\pm$ 0.21	49.65 $\pm$ 1.49	51.69 $\pm$ 1.05	48.82 $\pm$ 0.13 (0.1)
	SEM	36.41 $\pm$ 0.57	53.35 $\pm$ 0.29	52.38 $\pm$ 0.33	50.62 $\pm$ 1.89	51.63 $\pm$ 0.66	48.88 $\pm$ 0.39 (0.2)
	TYPO	36.62 $\pm$ 0.37	53.22 $\pm$ 0.54	52.58 $\pm$ 0.34	50.09 $\pm$ 0.86	51.40 $\pm$ 0.35	48.78 $\pm$ 0.11 (-0.1)
	-	33.32	52.35	51.36	50.57	50.14	47.55

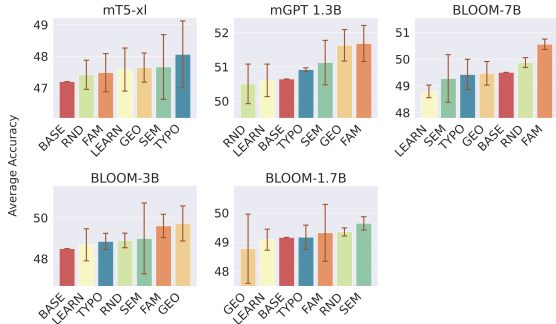


Figure 2: Average performance on unseen languages (not part of any instruction-tuning subset).

7B, only two of the five subsets improve over the base. This reveals a non-trivial interaction between pretraining corpus composition and subset selection: a strategy effective for one model’s pretraining distribution may be harmful for another’s.

3.4 Model Size Amplifies Differences—But Not Uniformly

Figure 3 shows how language subset performance evolves across BLOOM-1.7B, 3B, and 7B. While all instruction-tuned models surpass the base, the random baseline consistently ranks lowest among tuned models. GEO excels at 3B and 7B, whereas FAM starts strong at 1.7B but diminishes at scale.

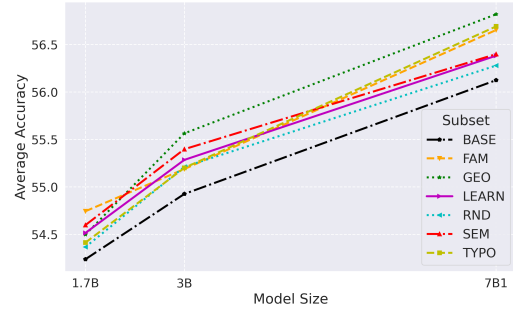


Figure 3: Average zero-shot performance across BLOOM model sizes by language subset. Strategy rankings shift with scale.

TYPO improves progressively with scale, reaching second place at 7B. Importantly, **the ranking of strategies changes across model sizes**: there is no strategy that is uniformly best as scale increases.

3.5 Monolingual Tuning Remains Competitive

In a Vietnamese case study (Figure 4), monolingual tuning (67k Vietnamese examples) ranks first for mT5 and second for mGPT on average across XNLI and XCOPA—with no statistically significant difference from the best multilingual subset (GEO) for mGPT. For BLOOM, the monolingual case consistently underperforms. This partially undermines the assumption that multilingual sub-

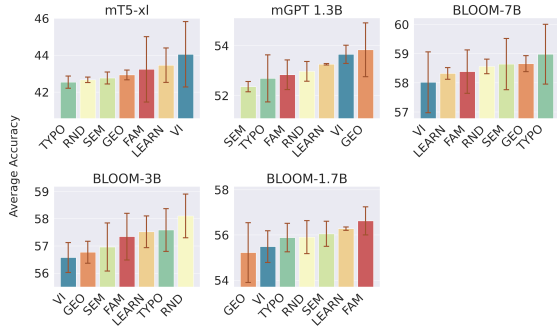


Figure 4: Monolingual (VI) vs. multilingual instruction tuning for Vietnamese on XNLI and XCOPA. Monolingual is competitive for mT5 and mGPT but not BLOOM.

sets are always preferable, and shows that the right choice is again model-dependent.

3.6 Training–Evaluation Language Overlap

Table 3 shows, for XNLI, how much overlap exists between each model’s pretraining languages (M), each training subset (S), and the evaluation languages (T). BLOOM stands out: its pretraining corpus covers only 8 of the 15 XNLI languages, and it has the smallest three-way overlaps of all models. This matches Table 2, where most informed subsets score significantly below RND on XNLI for BLOOM-7B.

mGPT instead covers 14 of the 15 XNLI languages, and the same informed subsets score significantly higher there. We find that overlap alone does not decide the outcomes, however: GEO improves XStoryCloze significantly on both BLOOM-7B and mGPT while sharing only 2 of its 11 evaluation languages. Since benchmark language sets themselves barely intersect (XNLI covers 15 languages, XWinograd 6), no single subset can align with every task.

4 Discussion and Implications

Why no optimal set exists. Three compounding factors explain the absence of a universal winner. (1) *Pretraining corpus mismatch*: languages that complement one model’s pretraining gaps differ across mGPT (61 languages), BLOOM (59, different sources), and mT5 (101). We quantify this: for BLOOM, a language’s accuracy closely tracks its share of the pretraining corpus, both before tuning (pooled Spearman $\rho=0.55\text{--}0.63$, $p<10^{-4}$) and after ($\rho=0.58\text{--}0.65$). For mT5, whose sampling flat-

Table 3: Language intersections for XNLI: counts of shared languages between the model’s pretraining corpus (M), the instruction tuning language subset (S), and the task’s evaluation languages (T). Bloom = BLOOM-7B.

Model	Subset	M-S-T	S-T	M-T	M-S
mT5-xl	ALL	13/15	13/15	15/15	50/52
mGPT	ALL	12/15	13/15	14/15	39/52
mT5-xl	RND	9/15	9/15	15/15	31/32
mT5-xl	TYPO	9/15	9/15	15/15	20/21
Bloom	ALL	8/15	13/15	8/15	18/46
mGPT	RND	8/15	9/15	14/15	24/32
mGPT	TYPO	8/15	9/15	14/15	18/21
mT5-xl	FAM	7/15	7/15	15/15	21/22
Bloom	RND	6/15	9/15	8/15	13/32
Bloom	TYPO	6/15	9/15	8/15	8/21
mGPT	FAM	6/15	7/15	14/15	20/22
Bloom	FAM	4/15	7/15	8/15	8/22
mT5-xl	GEO	4/15	4/15	15/15	15/16
mGPT	GEO	4/15	4/15	14/15	14/16
Bloom	GEO	4/15	4/15	8/15	8/16
mT5-xl	LEARN	3/15	3/15	15/15	18/20
mT5-xl	SEM	3/15	3/15	15/15	15/16
mGPT	LEARN	3/15	3/15	14/15	15/20
mGPT	SEM	3/15	3/15	14/15	11/16
Bloom	SEM	3/15	3/15	8/15	5/16
Bloom	LEARN	2/15	3/15	8/15	5/20

tens language proportions, the correlation vanishes ($\rho=-0.04$, n.s.). Yet a subset’s aggregate pretraining share does not predict tuned performance (n.s. in four of five models). Hence, pretraining explains where models perform well, but not which subset to tune on. (2) *Task–language alignment* (§3.6): high subset overlap with XNLI’s 15 languages does not imply overlap with XWinograd’s 6 or XCOPA’s 11, making universal optimality structurally impossible. (3) *Seed sensitivity*: three-seed confidence intervals are wide enough that strategy differences often fall within noise. These factors are not unique to our setting: Shimabucoro et al. (2025) reach the same conclusion at larger scale (up to 35B parameters), finding CLT dynamics vary jointly with task type, model size, and training regime.

Methodological warning. Reporting single aggregate scores across tasks and models in MIT can be deeply misleading. A strategy that appears superior in aggregate may underperform significantly on specific tasks: FAM achieves the best average for mT5-xl but scores worst on XNLI; GEO leads for mGPT overall but shows negative effects on BLOOM for the same task. Strikingly, the base non-tuned model scores best on PAWS-X for both BLOOM-7B and mGPT, suggesting that MIT can *harm* performance on certain task types.

We urge the community to always report per-task and per-model breakdowns with confidence intervals, and to avoid over-generalizing from a limited set of benchmarks or model families. Additionally, the zero-shot multiple-choice format of standard evaluation harnesses (Gao et al., 2023) has been critiqued for misaligning with real-world model use (Lyu et al., 2024), which may further explain cross-task variability.

Practical takeaway. Despite the absence of a universal optimum, linguistically-informed selection—particularly GEO and TYPO—*tends* to be a safer default than random selection, especially at larger model scales. For practitioners without resources to run full ablations, GEO-based selection via URIEL features (Littell et al., 2017) provides a compute-free heuristic that beats random on average. Alternatively, representation-space approaches such as LangGPS (Ye et al., 2026) offer a data-driven signal when model access is available. The key practical insight for dataset curators is to target 10–15 linguistically diverse languages rather than maximising language count: more languages is not better under a fixed computational budget.

5 Conclusion

We set out to find the optimal set of languages for multilingual instruction tuning. In our fixed-budget setting, we did not find one. Linguistically-informed language selection generally outperforms random selection on average, but no tested strategy is consistently best across tasks, model families, and evaluation settings. The curse of multilinguality reasserts itself beyond approximately 14 languages under a fixed computational budget; cross-lingual generalization to unseen languages is highly model-dependent; monolingual tuning can be competitive with multilingual approaches; and strategy rankings shift with model scale. Taken together, these negative findings caution against simple universal prescriptions for MIT data curation. They also highlight the need for task- and model-specific evaluation with proper statistical reporting—a standard the field has not yet widely adopted.

Limitations

Several limitations bear on the generalisability of our findings. First, we rely exclusively on LoRA (Hu et al., 2021) as the PEFT method; other fine-tuning approaches may yield different sensitivity to language subset choice. Second, Bactrian-

X (Li et al., 2023) consists largely of automatically translated English instructions, potentially introducing translation artefacts that confound language-specific effects. To gain more insights, we scored each language’s translation quality as the mean LaBSE (Feng et al., 2020) similarity between 500 English source instructions and their translations. We find that quality does not significantly correlate with per-language performance (Spearman $\rho=0.09-0.24$, n.s., vs. 0.58 for pretraining share over the same languages). Furthermore, the selection strategies barely differ in the mean quality of the languages they pick (0.862–0.877). Translation quality is therefore unlikely to explain our negative result, although LaBSE is itself an imperfect proxy.

In addition, all experiments are capped at 7B parameters due to resource constraints; behaviour at larger scales remains unknown. Fourth, only three random seeds per configuration limit statistical power, leaving confidence intervals wider than desirable—a key reason many strategy comparisons fail to reach significance. Fifth, language selection is constrained to the 52 Bactrian-X languages; a broader or different pool could yield different subsets.

Finally, the zero-shot multiple-choice evaluation format may not reflect instruction-following quality in open-ended generation settings (Lyu et al., 2024; Biderman et al., 2024), which is the more relevant real-world use case for instruction-tuned models. However, the models studied here generate outputs of too low quality for a meaningful open-ended evaluation, as we observed in preliminary LLM-as-a-judge experiments.

Acknowledgments

This work has been supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) as part of the project “Automatic Learning of Procedural Language from Natural Language Instructions for Intelligent Assistance” with the number 121C132. We also gratefully acknowledge KUIS AI Lab for providing computational support.

References

- Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sid Black, Jordan Clive, Anthony DiPofi, Julien Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, and 10 others. 2024. [Lessons from the trenches on reproducible evaluation of language models](#). *ArXiv*, abs/2405.14782.
- Pinzhen Chen, Shaoxiong Ji, Nikolay Bogoychev, Barry Haddow, and Kenneth Heafield. 2023a. Monolingual or multilingual instruction tuning: Which makes a better alpaca. *arXiv preprint arXiv:2309.08958*.
- Yiyi Chen, Russa Biswas, and Johannes Bjerva. 2023b. [Colecx2Lang: Language embeddings from semantic typology](#). In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 673–684, Tórshavn, Faroe Islands. University of Tartu Library.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Unsupervised cross-lingual representation learning at scale](#). In *Annual Meeting of the Association for Computational Linguistics*.
- Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. [XNLI: Evaluating cross-lingual sentence representations](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Matthew Cer, N. Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT Sentence Embedding. In *Annual Meeting of the Association for Computational Linguistics*.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2023. [A framework for few-shot language model evaluation](#). Zenodo. Version v0.4.0, DOI: 10.5281/zenodo.10256836.
- Harald Hammarström, Robert Forkel, Martin Haspelmath, and Sebastian Bank. 2023. [Glottolog database 4.8](#). Available online at <https://glottolog.org>.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. [The state and fate of linguistic diversity and inclusion in the NLP world](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.
- Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023. [Bactrian-X: A multilingual replicable instruction-following model with low-rank adaptation](#). *Preprint*, arXiv:2305.15011.
- Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O’Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, and 2 others. 2021. [Few-shot learning with multilingual language models](#). *CoRR*, abs/2112.10668.
- Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *The 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy.
- Patrick Littell, David R Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 8–14.
- Chenyang Lyu, Minghao Wu, and Alham Aji. 2024. [Beyond probabilities: Unveiling the misalignment in evaluating large language models](#). In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 109–131, Bangkok, Thailand. Association for Computational Linguistics.
- Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Copenhagen, Denmark.
- Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. [Crosslingual generalization through multitask finetuning](#). *Preprint*, arXiv:2211.01786.
- Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020.

- XCOPA: A multilingual dataset for causal common-sense reasoning.** In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.
- Teven Le Scao, Angela Fan, Christopher Akiki, Elie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, and 30 others. 2022. **BLOOM: A 176b-parameter open-access multilingual language model.** *CoRR*, abs/2211.05100.
- Uri Shaham, Jonathan Herzig, Roei Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2024. **Multilingual instruction tuning with just a pinch of multilinguality.** *ArXiv*, abs/2401.01854.
- Luís Shimabucoro, Ahmet Üstün, Marzieh Fadaee, and Sebastian Ruder. 2025. A post-trainer’s guide to multilingual training data: Uncovering cross-lingual transfer dynamics. *arXiv preprint arXiv:2504.16677*.
- Oleh Shliakhko, Alena Fenogenova, Maria Tikhonova, Anastasia Kozlova, Vladislav Mikhailov, and Tatiana Shavrina. 2024. **mGPT: Few-shot learners go multilingual.** *Transactions of the Association for Computational Linguistics*, 12:58–79.
- Alexey Tikhonov and Max Ryabinin. 2021. **It’s all in the heads: Using attention heads as a baseline for cross-lingual transfer in commonsense reasoning.** *Preprint*, arXiv:2106.12066.
- Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. **UDapter: Language adaptation for truly Universal Dependency parsing.** In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.
- Alexander Arno Weber, Klaudia Thellmann, Jan Ebert, Nicolas Flores-Herr, Jens Lehmann, Michael Fromm, and Mehdi Ali. 2024. **Investigating multilingual instruction-tuning: Do polyglot models demand for multilingual instructions?** In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20829–20855, Miami, Florida, USA. Association for Computational Linguistics.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. **mT5: A massively multilingual pre-trained text-to-text transformer.** In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. **PAWS-X: A cross-lingual adversarial dataset for paraphrase identification.** In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.
- Yangfan Ye, Xiaocheng Feng, Xiachong Feng, Lei Huang, Weitao Ma, Qichen Hong, Yunfei Lu, Duyu Tang, Dandan Tu, and Bing Qin. 2026. **LangGPS: Language separability guided data pre-selection for joint multilingual instruction tuning.** In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Short-Context Gains, Long-Context Drift: Negative Results for Token-Conditioned RoPE

Allan Kazakov

Bahçeşehir University

allan.kazakov@bahcesehir.edu.tr

Abstract

RoPE is a standard positional encoding for modern language models, but its fixed rotations can fail to distinguish positions and tokens in long contexts. Content-aware rotary methods such as CARoPE suggest that adapting rotary geometry to token content can improve language modeling. We report a negative-result study of four token-conditioned RoPE variants for language-model pretraining. DiPE adds token-dependent endpoint phase offsets; Gated DiPE bounds these offsets with learned gates; HSA-RoPE and PPA-RoPE instead modulate accumulated rotary frequency increments at the head and rotary-pair levels. Although these methods produce promising short-context gains, their improvements do not scale reliably. DiPE improves substantially at sequence length 1024 but weakens with larger models and longer contexts, while Gated DiPE fails to recover the lost benefit. HSA-RoPE and PPA-RoPE outperform RoPE at all tested sequence lengths, but their advantage shrinks with longer contexts and they fall substantially behind CARoPE at 8192 tokens.

1 Introduction

Rotary position embeddings (RoPE) encode relative position by rotating query and key vectors with fixed, dimension-specific frequencies (Su et al., 2021). This design is simple and widely used in decoder-only language models (Touvron et al., 2023), but its geometry is independent of token content. Recent work shows that fixed RoPE can fail to distinguish positions and tokens in long contexts (Du et al., 2026), while CARoPE demonstrates that content-aware rotary frequencies can improve language modeling (Veisi et al., 2025). These results motivate a narrower question: when does injecting token content into rotary geometry help, and when does it damage the positional structure that RoPE provides? We study this question

through a negative-result investigation of token-conditioned RoPE variants. Our starting point is Dimension-wise Phase-offset Embeddings (DiPE), which adds token-dependent endpoint phase offsets to standard RoPE. DiPE is initially promising: in small-model training at sequence length 1024, it outperforms both RoPE and CARoPE. However, its gains weaken with larger models and longer contexts. A gated version, which bounds endpoint offsets with learned gates and phase caps, also fails to recover the lost long-context benefit. We then test two accumulated-frequency variants designed to preserve more of RoPE’s interval structure. Head-Scalar Accumulated RoPE (HSA-RoPE) predicts one bounded frequency scale per token and attention head, while Per-Pair Accumulated RoPE (PPA-RoPE) predicts one scale per token, head, and rotary pair. Both are initialized to exactly match RoPE. They outperform RoPE at all tested sequence lengths, but their relative advantage shrinks with context length: they beat CARoPE at 1024 tokens, become weaker on validation at 4096 tokens, and fall substantially behind CARoPE at 8192 tokens. Diagnostics suggest that the degradation is not caused by token conditioning itself, but by the relative phase geometry induced by each parameterization. Endpoint offsets can behave like content-dependent phase noise at long distances. HSA-RoPE learns a persistent slowdown of the rotary clock, producing large accumulated phase drift, while PPA-RoPE can misalign rotary frequency channels despite small average scale shifts. Thus, our results do not imply that content-aware rotary frequencies are inherently flawed: CARoPE remains a strong counterexample. Rather, they show that RoPE initialization and local bounds are not sufficient; successful token-conditioned rotary mechanisms must correct RoPE’s long-context weaknesses while keeping relative phases controlled.

2 Methods

We compare RoPE with four token-conditioned rotary variants under the same Transformer backbone (Vaswani et al., 2017). Let θ_i denote the standard RoPE frequency for rotary pair i , and let x_t be the token embedding at position t . Standard RoPE applies the phase

$$\phi_{t,i} = t\theta_i \quad (1)$$

to each query and key rotary pair. All methods below modify this phase while preserving the same attention, MLP, tokenizer, and training pipeline.

DiPE. Dimension-wise Phase-offset Embeddings add a token-conditioned endpoint offset to RoPE. A linear projection maps each token embedding to angles $\alpha_{t,h,i}$ for each attention head h and rotary pair i :

$$\phi_{t,h,i}^{\text{DiPE}} = t\theta_i + \alpha_{t,h,i}. \quad (2)$$

Thus, the relative phase between positions m and n becomes

$$(m - n)\theta_i + \alpha_{m,h,i} - \alpha_{n,h,i}. \quad (3)$$

This gives attention access to endpoint token content, but the added term is not tied to the interval between the two positions.

Gated DiPE. Gated DiPE uses the same endpoint-offset form, but bounds the learned offset with a phase cap and a learned per-head, per-pair gate:

$$\phi_{t,h,i}^{\text{Gated}} = t\theta_i + g_{h,i} \tilde{\alpha}_{t,h,i}, \quad (4)$$

where $\tilde{\alpha}_{t,h,i}$ is the capped DiPE angle and $g_{h,i}$ is a bounded learned gate. This variant tests whether allowing the model to control the strength of endpoint offsets is sufficient to recover stable long-context behavior.

HSA-RoPE. Head-Scalar Accumulated RoPE removes endpoint offsets and instead predicts one bounded positive frequency scale per token and attention head:

$$s_{t,h} = \exp(c \tanh(W_h x_t)). \quad (5)$$

The local rotary increment is then

$$\Delta\phi_{t,h,i} = \theta_i s_{t,h}, \quad (6)$$

and the final phase is accumulated over positions:

$$\phi_{t,h,i}^{\text{HSA}} = \sum_{u=1}^t \theta_i s_{u,h}. \quad (7)$$

When $s_{t,h} = 1$ for all tokens and heads, HSA-RoPE exactly recovers RoPE. Since the same scale is shared across all rotary pairs within a head, HSA-RoPE preserves the internal RoPE frequency ladder inside each head.

PPA-RoPE. Per-Pair Accumulated RoPE generalizes HSA-RoPE by predicting a separate bounded scale for every token, head, and rotary pair:

$$s_{t,h,i} = \exp(c \tanh(W_{h,i} x_t)). \quad (8)$$

Its phase is

$$\phi_{t,h,i}^{\text{PPA}} = \sum_{u=1}^t \theta_i s_{u,h,i}. \quad (9)$$

PPA-RoPE is more expressive than HSA-RoPE because each rotary frequency can adapt independently. However, this flexibility can also distort the relative alignment between frequency channels.

Initialization. HSA-RoPE and PPA-RoPE are initialized to exactly match RoPE by setting their scale projections to zero, giving $s = 1$ at initialization. DiPE starts close to RoPE through near-zero angle initialization, while Gated DiPE additionally starts with bounded gates. Therefore, performance differences arise from learned departures from the RoPE geometry rather than from different initial positional encodings.

3 Experimental Setup

We evaluate all methods in autoregressive decoder-only language-model pretraining, following the standard next-token prediction setting used by GPT-style models (Radford et al., 2019; Brown et al., 2020). Unless stated otherwise, models are trained on the same packed tokenized corpus and use the same tokenizer, transformer backbone, optimizer settings, training budget, and evaluation protocol. The corpus contains approximately 20B tokens from web, wiki, code, and math sources, with a two-stage mixture schedule: a broad 16B-token stage followed by a 4B-token code/math-upweighted stage. Additional data and training details are given in Appendix A.

Our main experiments use a small decoder-only transformer and vary the positional mechanism while keeping the rest of the architecture fixed. We compare baseline RoPE, CARoPE, HSA-RoPE, and PPA-RoPE at sequence lengths 1024, 4096, and 8192. Because endpoint DiPE and Gated DiPE

are already weaker than the accumulated-frequency variants at 4096 tokens, we reserve the 8192-token runs for HSA-RoPE and PPA-RoPE, since they are the most promising variants. Baseline RoPE and CARoPE use the same architecture and hyperparameters, differing only in the rotary mechanism. For HSA-RoPE and PPA-RoPE, we first run a max-log-scale sweep at sequence length 4096 and select the best validation settings, $c = 0.25$ for HSA-RoPE and $c = 0.1$ for PPA-RoPE. We then keep these values fixed for all sequence-length experiments. The sweep results are reported in Appendix C.

To test whether short-context gains survive model scaling, we additionally run a larger 24-layer model at sequence length 1024, with hidden size 1024, 16 attention heads, and MLP dimension 2816. The vocabulary size is fixed at 50,304. This larger-model study is used only as a scale ablation, not as part of the sequence-length scaling comparison.

We report final training loss, validation loss, and perplexity. For diagnostic analysis, we track learned frequency scales and phase differences relative to RoPE. For PPA-RoPE, we also report wrapped phase differences, which measure angular misalignment modulo 2π . These diagnostics are used to distinguish global rotary-clock drift from per-frequency angular misalignment.

4 Results

Sequence-length scaling. Table 1 shows the main scaling result. These final-checkpoint results reflect stable late-training behavior rather than a noisy endpoint: validation curves show the same ordering near convergence, with no reversal among the main methods (Appendix G). At 1024 tokens, all token-conditioned variants improve over RoPE, with PPA-RoPE giving the best validation loss. However, the relative ordering changes as context length increases. At 4096 tokens, HSA-RoPE and PPA-RoPE still improve over RoPE, but CARoPE becomes best on validation. At 8192 tokens, HSA-RoPE and PPA-RoPE remain slightly better than RoPE, but their gains shrink substantially, while CARoPE improves by a much larger margin. Thus, accumulated-frequency modulation is not automatically long-context robust: it helps over RoPE, but does not preserve its short-context advantage over CARoPE.

Gating endpoint offsets does not fix DiPE. At 4096 tokens, DiPE improves over RoPE, but Gated

Seq. len.	Method	Val loss	Val PPL
1024	RoPE	2.770	15.96
	CARoPE	2.769	15.94
	DiPE	2.739	15.48
	HSA-RoPE	2.734	15.39
	PPA-RoPE	2.724	15.24
4096	RoPE	2.670	14.44
	CARoPE	2.627	13.83
	DiPE	2.644	14.07
	Gated DiPE	2.666	14.38
	HSA-RoPE	2.635	13.95
	PPA-RoPE	2.634	13.93
8192	RoPE	2.638	13.99
	CARoPE	2.594	13.38
	HSA-RoPE	2.627	13.84
	PPA-RoPE	2.628	13.84

Table 1: Validation performance across sequence lengths. HSA-RoPE and PPA-RoPE outperform RoPE at all tested lengths, but their advantage shrinks with context length, while CARoPE becomes strongest at 4096 and 8192 tokens. Full training losses and perplexities are reported in Appendix B with training and validation curves in Appendix G.

DiPE does not: its validation loss is close to RoPE and worse than DiPE, HSA-RoPE, PPA-RoPE, and CARoPE. This suggests that the issue is not simply that endpoint offsets are too strong. Even when the model can bound them with learned gates and phase caps, the endpoint-offset parameterization does not recover robust long-context behavior.

Model scaling weakens the short-context story.

We also test a larger 24-layer model at sequence length 1024. In this setting, DiPE and HSA-RoPE achieve lower training loss than CARoPE, but CARoPE obtains the best validation loss. HSA-RoPE is especially illustrative: it reaches much lower training loss, yet validates slightly worse than CARoPE. This indicates that lower pretraining loss from token-conditioned rotary modulation does not necessarily imply better generalization. We report the full larger-model ablation in Appendix D.

Angular diagnostics. Table 2 summarizes final 8192-token diagnostics for HSA-RoPE and PPA-RoPE. For diagnostics, we compare each learned rotary phase to the corresponding RoPE phase and report the absolute difference. We report both the mean and the 95th percentile (p95), computed over the final evaluation batch, sequence positions, attention heads, and rotary pairs. For PPA-RoPE, we also report the wrapped phase difference, computed as the absolute angular difference modulo 2π .

HSA-RoPE learns a persistent slowdown of the

Method	Scale mean	Scale std.	Phase diff. mean	Phase diff. p95	Wrapped diff. mean	Wrapped diff. p95
HSA-RoPE	0.911	0.154	52.08	293.8	—	—
PPA-RoPE	0.978	0.033	5.75	28.80	0.909	2.789

Table 2: Final angular diagnostics at sequence length 8192. HSA-RoPE did not log wrapped phase metrics, so we report unwrapped phase drift for HSA-RoPE and both unwrapped and wrapped phase differences for PPA-RoPE. Diagnostics for 1024 and 4096 tokens are given in Appendix E.

rotary clock: its mean scale is 0.911, and its accumulated phase drift is large, with mean absolute phase delta 52.08 and p95 293.8. PPA-RoPE has much smaller average scale deviation, but its wrapped phase error is still large: the p95 wrapped phase delta is 2.789 radians. These diagnostics support two different failure modes. HSA-RoPE deforms RoPE mainly through accumulated global drift, whereas PPA-RoPE can misalign individual frequency channels despite smaller average scale changes. The diagnostic results for other sequence lengths are described in Appendix E.

5 Discussion

Our experiments suggest that content-aware rotary modulation is useful but fragile. The strongest short-context results come from mechanisms that allow token content to alter rotary phases or frequencies, but these gains do not reliably transfer to longer contexts or larger models. This matches prior length-extrapolation work showing that positional mechanisms should be evaluated at the target context length rather than only at short or training lengths (Press et al., 2022; Chen et al., 2023; Peng et al., 2023; Ding et al., 2024).

The comparison with CARoPE is important. CARoPE shows that content-aware rotary frequencies can scale well, so our results should not be interpreted as evidence against token-conditioned RoPE in general. Instead, they show that the parameterization of the content-conditioned rotation matters. RoPE initialization ensures that a method starts from a stable geometry, and local bounds limit each token’s immediate effect, but neither prevents harmful cumulative behavior. HSA-RoPE illustrates this through global rotary-clock drift, while PPA-RoPE illustrates it through per-frequency angular misalignment.

These findings also clarify why endpoint phase offsets are difficult to stabilize. DiPE modifies the phase assigned to each endpoint token rather than the frequency path between positions. This can help at short context, where the offset acts as an

additional content signal, but it does not preserve a clean interval structure. Gating and phase caps reduce the magnitude of the perturbation, but they do not change this endpoint-based parameterization.

A practical takeaway is that content-conditioned rotary methods should be evaluated at the target context length, not only at short context. They should also be diagnosed geometrically, since training loss alone can be misleading: in our larger-model ablation, HSA-RoPE obtains the lowest training loss but does not achieve the best validation loss. Future designs may need constraints that directly control accumulated relative phase drift, preserve alignment across rotary frequency channels, or regularize deviations from a well-behaved adaptive frequency path.

6 Conclusion

We presented a negative-result study of token-conditioned RoPE variants for language-model pre-training. DiPE, Gated DiPE, HSA-RoPE, and PPA-RoPE all test plausible ways to make rotary geometry depend on token content, and several improve over RoPE at short or moderate context. However, endpoint offsets do not scale reliably, gating does not fix them, and accumulated-frequency variants lose their advantage over CARoPE at longer context. These failures should not discourage content-aware positional embeddings. Instead, they suggest that improving RoPE remains a promising direction, but successful methods must adapt rotary geometry while controlling the relative phases induced by that adaptation. More broadly, our results support continued work on positional mechanisms that are not only position-dependent, but also token-dependent in a geometrically constrained way.

Code Availability

Code for all models, training scripts, and diagnostics is available at <https://github.com/AllanK24/dipe-hsa-ppa>.

Limitations

Our study is limited to decoder-only language-model pretraining on one corpus and a small number of model scales. We do not run endpoint DiPE or Gated DiPE at 8192 tokens because they are already weaker than the accumulated-frequency variants at 4096 tokens. HSA-RoPE and PPA-RoPE use max-log-scale values selected from a 4096-token sweep; although limited 1024-token checks favored the same settings, we did not retune these bounds separately for every sequence length or model scale. Finally, our diagnostics focus on rotary phase geometry; they explain observed failures but do not fully characterize how these phase changes interact with downstream attention patterns.

References

- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. [Extending context window of large language models via positional interpolation](#). *arXiv preprint arXiv:2306.15595*.
- Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. [LongRoPE: Extending LLM context window beyond 2 million tokens](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11091–11104. PMLR.
- Yufeng Du, Phillip Harris, Minyang Tian, Eliu A. Huerta, Srikanth Ronanki, Subendhu Rongali, Aram Galstyan, and Hao Peng. 2026. [RoPE distinguishes neither positions nor tokens in long contexts, provably](#). *arXiv preprint arXiv:2605.15514*.
- Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. [YaRN: Efficient context window extension of large language models](#). *arXiv preprint arXiv:2309.00071*.
- Ofir Press, Noah A. Smith, and Mike Lewis. 2022. [Train short, test long: Attention with linear biases enables input length extrapolation](#). In *International Conference on Learning Representations*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). *OpenAI technical report*.
- Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. [RoFormer: Enhanced transformer with rotary position embedding](#). *arXiv preprint arXiv:2104.09864*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothee Lacroix, Baptiste Roziere, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [LLaMA: Open and efficient foundation language models](#). *arXiv preprint arXiv:2302.13971*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30.
- Ali Veisi, Delaram Fartoot, and Hamidreza Amirzadeh. 2025. [Context-aware rotary position embedding](#). *arXiv preprint arXiv:2507.23083*.

A Additional Experimental Details

All experiments use decoder-only Transformer language models trained with next-token prediction. Within each comparison, models use the same tokenizer, packed training corpus, optimizer settings, training budget, and evaluation protocol; only the positional mechanism is varied. The vocabulary size is fixed at 50,304. The small-model experiments use a 12-layer Transformer with hidden size 768, 12 attention heads, and MLP dimension 2048. We evaluate this model at sequence lengths 1024, 4096, and 8192. The larger-model ablation uses a 24-layer Transformer with hidden size 1024, 16 attention heads, and MLP dimension 2816, evaluated only at sequence length 1024. The training corpus contains approximately 20B tokens from web, wiki, code, and math sources. Specifically, we use HuggingFaceFW/fineweb_edu_100BT-shuffled, the English subset of HuggingFaceFW/finewiki, HuggingFaceTB/finemath with configuration finemath-4plus, and the Python subset of bigcode/the-stack-dedup. We use a two-stage mixture schedule: a broad 16B-token stage followed by a 4B-token stage with code/math upweighting. Most core experiments are trained with distributed data parallelism on 8 H100 GPUs. We evaluate validation loss on the same held-out

mixture for all methods within each comparison. The number of tokens per optimizer update is fixed at 524,288 across model configurations. As a result, the global batch size varies with sequence length. All runs use learning rate 2×10^{-4} , minimum learning rate 2×10^{-5} , cosine learning-rate decay, 1,000 warmup steps, weight decay 0.1, gradient clipping at 1.0, and random seed 42.

B Full Sequence-Length Results

Table 3 reports the full training and validation results corresponding to Table 1. The main paper reports only validation metrics for compactness.

C Max-Log-Scale Sweeps

HSA-RoPE and PPA-RoPE use a bounded log-scale parameter c that controls the maximum local frequency deformation. We tune this parameter at sequence length 4096 and use the selected values for all HSA-RoPE and PPA-RoPE experiments, including the sequence-length scaling study and the larger-model ablation. Table 4 shows that HSA-RoPE performs best with $c = 0.25$, while PPA-RoPE performs best with the tighter bound $c = 0.1$.

The sweep also shows that PPA-RoPE is more sensitive to the scale bound than HSA-RoPE. Increasing PPA-RoPE from $c = 0.1$ to $c = 0.15$ or $c = 0.25$ substantially worsens validation loss, consistent with the interpretation that per-pair frequency freedom can distort the rotary frequency ladder. We also ran limited checks at sequence length 1024, testing $c = 0.15$ and $c = 0.30$ for HSA-RoPE and $c = 0.05$ and $c = 0.20$ for PPA-RoPE. These checks did not improve over the values selected at 4096, so we use the same settings throughout.

D Larger-Model Scale Ablation

Table 5 reports the 24-layer, sequence-length-1024 ablation. DiPE and HSA-RoPE improve training loss relative to CARoPE, but CARoPE obtains the best validation loss. This supports the main-paper claim that token-conditioned rotary gains in training loss do not necessarily translate to better validation performance.

E Additional Angular Diagnostics

Table 6 reports final angular diagnostics for HSA-RoPE and PPA-RoPE across sequence lengths. For

each method, phase differences are measured relative to the corresponding RoPE phase. Mean and p95 statistics are computed over the final evaluation batch, sequence positions, attention heads, and rotary pairs. HSA-RoPE did not log wrapped phase differences, so only unwrapped phase differences are reported for HSA-RoPE.

The diagnostics show that HSA-RoPE accumulates increasingly large phase drift as sequence length grows: mean absolute phase difference rises from 6.025 at 1024 tokens to 52.08 at 8192 tokens. PPA-RoPE has smaller unwrapped drift, but its wrapped phase difference becomes large by 4096 tokens and remains high at 8192 tokens. This supports the interpretation that HSA-RoPE fails mainly through global rotary-clock drift, while PPA-RoPE fails through per-frequency angular misalignment.

F Parameter Counts

Table 7 reports total parameter counts. All methods keep the Transformer backbone fixed and differ only in the positional mechanism. The additional parameters introduced by the token-conditioned rotary variants are small relative to the full model size. For example, in the small model, HSA-RoPE and CARoPE add only 9.2K parameters over RoPE, while DiPE and PPA-RoPE add about 0.3M parameters.

G Training and Validation Curves

Figure 1 shows training and validation loss curves for the 8192-token setting. The final-checkpoint results in Table 3 are consistent with the late-training trajectories. CARoPE remains consistently below RoPE, HSA-RoPE, and PPA-RoPE, while HSA-RoPE and PPA-RoPE remain close to each other and only modestly better than RoPE near convergence. This supports using final training and validation losses as the main comparison metrics.

Seq. len.	Method	Train loss	Train PPL	Val loss	Val PPL
1024	RoPE	2.685329	14.663021	2.770166	15.961282
1024	CARoPE	2.685213	14.661318	2.768793	15.939384
1024	DiPE	2.651714	14.178326	2.739253	15.475425
1024	HSA-RoPE	2.645509	14.090617	2.733593	15.388072
1024	PPA-RoPE	2.634428	13.935340	2.723719	15.236888
4096	RoPE	3.182270	24.101403	2.669760	14.436501
4096	CARoPE	3.144775	23.214455	2.627078	13.833295
4096	DiPE	3.151549	23.372245	2.643735	14.065646
4096	Gated DiPE	3.184689	24.159778	2.665644	14.377202
4096	HSA-RoPE	3.140824	23.122920	2.635327	13.947866
4096	PPA-RoPE	3.144899	23.217336	2.633718	13.925453
8192	RoPE	2.513150	12.343751	2.638428	13.991198
8192	CARoPE	2.478648	11.925133	2.593681	13.378933
8192	HSA-RoPE	2.501863	12.205205	2.627347	13.837014
8192	PPA-RoPE	2.506641	12.263672	2.627726	13.842258

Table 3: Full sequence-length scaling results. Endpoint DiPE and Gated DiPE were not run at 8192 tokens because they were already weaker than the accumulated-frequency variants at 4096 tokens.

Method	c	Train loss	Train PPL	Val loss	Val PPL
HSA-RoPE	0.10	3.163375	23.650270	2.653924	14.209693
HSA-RoPE	0.15	3.158291	23.530356	2.649613	14.148564
HSA-RoPE	0.25	3.140824	23.122920	2.635327	13.947866
HSA-RoPE	0.30	3.141033	23.127736	2.641458	14.033652
PPA-RoPE	0.10	3.144899	23.217336	2.633718	13.925453
PPA-RoPE	0.15	3.159697	23.563459	2.649472	14.146564
PPA-RoPE	0.25	3.159797	23.565816	2.651016	14.168431

Table 4: Max-log-scale sweep at sequence length 4096. We select $c = 0.25$ for HSA-RoPE and $c = 0.1$ for PPA-RoPE, then keep these values fixed in all reported HSA-RoPE and PPA-RoPE experiments.

Method	Train loss	Train PPL	Val loss	Val PPL
RoPE	2.323361	10.209929	2.442804	11.505257
CARoPE	2.308816	10.062504	2.435992	11.427148
DiPE	2.307163	10.045883	2.439677	11.469333
HSA-RoPE	2.224412	9.248047	2.440316	11.476667

Table 5: Larger-model ablation at sequence length 1024. PPA-RoPE did not finish because TorchInductor compilation errors made the run prohibitively slow.

Seq. len.	Method	Scale mean	Scale std.	Phase mean	Phase p95	Wrapped mean	Wrapped p95
1024	HSA-RoPE	0.9543	0.1769	6.025	34.48	—	—
1024	PPA-RoPE	0.9796	0.0325	0.9018	4.678	0.4288	2.165
4096	HSA-RoPE	0.9294	0.1405	19.87	125.9	—	—
4096	PPA-RoPE	0.9555	0.0636	5.605	32.83	0.9195	2.800
8192	HSA-RoPE	0.9112	0.1538	52.08	293.8	—	—
8192	PPA-RoPE	0.9783	0.0332	5.752	28.80	0.9085	2.789

Table 6: Final angular diagnostics across sequence lengths. Phase columns report absolute phase difference from RoPE. Wrapped columns report absolute angular difference modulo 2π .

Model size	Method	Total params	Extra over RoPE
Small	RoPE	123,587,328	–
	CARoPE	123,596,556	9,228
	DiPE	123,882,240	294,912
	Gated DiPE	123,882,624	295,296
	HSA-RoPE	123,596,556	9,228
	PPA-RoPE	123,882,624	295,296
Large	RoPE	359,842,816	–
	CARoPE	359,859,216	16,400
	DiPE	360,367,104	524,288
	HSA-RoPE	359,859,216	16,400
	PPA-RoPE	360,367,616	524,800

Table 7: Total parameter counts and additional parameters over the RoPE baseline. All methods keep the Transformer backbone fixed.



Figure 1: Training and validation loss curves at sequence length 8192. Training loss is smoothed with a centered rolling window of 31 steps; validation loss is shown for steps after 34K. CARoPE remains consistently below the other methods near convergence, while HSA-RoPE and PPA-RoPE provide only modest improvements over RoPE.

When Gradient Importance Lies: Adaptive LoRA Rank Allocation Fails Under GRPO

Yash Ganpat Sawant
Independent AI Researcher
sawanttyash13@gmail.com

Abstract

Adaptive rank allocation for LoRA—allocating more parameters to important layers and fewer to unimportant ones—consistently improves efficiency under supervised fine-tuning (SFT). We test whether this success transfers to reinforcement learning, specifically Group Relative Policy Optimization (GRPO). Using gradient-magnitude profiling on Qwen 2.5 1.5B with GSM8K, we find that, *in our setting*, it does not: proportional rank allocation *degrades* accuracy by 4.5 points compared to uniform allocation (70.0% vs. 74.5%), despite using identical parameter budgets. We identify two mechanisms behind this failure. First, the gradient landscape under GRPO is fundamentally flatter than under SFT—the max-to-min layer importance ratio is only $2.17\times$, whereas the layer concentration reported by Shi et al. (2024) for SFT (top 30% of layers carrying $>80\%$ of the gradient signal) implies a max/min ratio well above $10\times$. All layers carry meaningful gradient signal; none are truly idle. Second, we observe a *gradient amplification effect*: non-uniform allocation widens the importance spread from $2.17\times$ to $3.00\times$, creating a positive feedback loop where high-rank layers absorb more gradient while low-rank layers are progressively silenced. A random-allocation control yields the same amplification ($r = 0.972$ correlation between assigned rank and resulting gradient share), indicating that rank causally determines gradient importance rather than the reverse. The negative result is single-seed and single-task; we present it as preliminary evidence that gradient importance does not predict capacity requirements under RL, and that naïve transfer of SFT-era rank allocation strategies to alignment training should be evaluated cautiously.

1 Introduction

Parameter-efficient fine-tuning via Low-Rank Adaptation (Hu et al., 2022) is the standard ap-

proach for adapting large language models. LoRA decomposes weight updates into low-rank matrices $\Delta W = BA$ with uniform rank r across all layers. A line of follow-up work has challenged this uniform assumption. AdaLoRA (Zhang et al., 2023) dynamically prunes singular values during training based on importance scores. GoRA (He et al., 2025) allocates rank proportionally to gradient-weight products at initialization. Aletheia (Saket, 2026) selects layers via lightweight gradient probes. ILA (Shi et al., 2024) shows that only 10–30% of layers carry the bulk of the gradient signal for alignment under SFT, with the top 30% carrying $>80\%$ of total magnitude. These methods report meaningful efficiency gains, supporting the picture that **gradient importance correlates with capacity requirements under supervised objectives**.

A natural question follows: does this correlation hold under reinforcement learning? RL-based alignment methods like GRPO (Shao et al., 2024) optimize an advantage-weighted policy gradient with sparse, often binary reward, rather than a dense per-token cross-entropy loss. Theoretical work suggests RL gradients concentrate differently than SFT gradients (Young, 2026), motivating the hypothesis that rank allocation strategies should differ.

We report a negative result. In our setting—Qwen 2.5 1.5B fine-tuned on GSM8K with GRPO—applying the same gradient-magnitude profiling and proportional allocation recipe that helps SFT *hurts* GRPO accuracy by 4.5 points relative to uniform allocation at an identical parameter budget. A random control degrades further, but uniform allocation outperforms both. The result is preliminary (one model, one dataset, single seed, $n = 200$ eval), but the mechanism we expose is robust across our four configurations and we believe it is worth surfacing.

We make three observations:

1. **Flat gradient landscape.** Under GRPO on this task, layer importance is distributed far more uniformly than what Shi et al. (2024) report for SFT (max/min ratio $2.17\times$ vs. an implied $>10\times$). All layers carry load.
2. **Gradient amplification.** Non-uniform allocation widens the importance spread from $2.17\times$ to $3.00\times$. High-rank layers absorb more gradient; low-rank layers are silenced. Rank thus causally determines the observed gradient distribution.
3. **Generalization gap.** Models train equally well across allocations (Fig. 2), but the proportional and random configurations fail to transfer to the held-out test set. The damage from rank reallocation appears only at evaluation time.

We frame these findings as a case study in why SFT-era efficiency tricks should not be transferred to RL without empirical validation.

2 Method

2.1 GRPO with LoRA

GRPO generates K completions per prompt, computes rewards, and normalizes advantages within the group:

$$\hat{A}_i = \frac{r_i - \mu_{\text{group}}}{\sigma_{\text{group}}} \quad (1)$$

Combined with LoRA, gradients flow only through the adapter parameters $B^{(l)}, A^{(l)}$ at each layer l .

2.2 Reward Sensitivity Profiling

We define the reward sensitivity score for layer l as the mean over T training steps of the summed adapter gradient norm:

$$S(l) = \frac{1}{T} \sum_{t=1}^T \sum_{m \in \mathcal{M}^{(l)}} \left\| \nabla_{\theta_m^{(l)}} \mathcal{L}_t \right\|_2 \quad (2)$$

where $\mathcal{M}^{(l)}$ is the set of target modules at layer l (q/k/v/o/up/down/gate projections). Equation 2 aggregates over modules within a layer; the per-module decomposition we use later for the attention vs. FFN analysis (§3.5) is computed separately and reported in Appendix A.

2.3 Rank Allocation

Given a total rank budget $R_{\text{total}} = L \times r_{\text{uniform}}$, we first normalize the sensitivity score, $\hat{s}(l) = S(l) / \sum_{l'} S(l')$, and then allocate per-layer rank

$$r^{(l)} = \text{clip}(\text{round}(\hat{s}(l) \cdot R_{\text{total}}), r_{\min}, r_{\max}) \quad (3)$$

with $r_{\min} = 4, r_{\max} = 64$, rounded to multiples of 4. As a control we also evaluate *random* allocation: a permutation of the proportional rank vector across layers, removing any link to $S(l)$ while preserving the rank distribution and budget. A *reduced* 70% variant scales the proportional vector to 70% of the uniform budget ($r \in [12, 31]$, 25.8M trainable parameters vs. 36.9M for the same-budget conditions) to probe whether smaller models can match uniform performance.

3 Experiments

3.1 Setup

Model: Qwen/Qwen2.5-1.5B-Instruct (28 transformer layers). **Dataset:** GSM8K (Cobbe et al., 2021) with structured XML output (`<think><answer>`). **Rewards:** Format compliance (1.0 for correctly closed tags) and answer correctness (1.0 for the correct numerical answer). **LoRA:** Applied to all seven projection modules per layer (196 adapters total). Uniform baseline: $r = 32$, giving 36.9M trainable parameters. Across every configuration we scale the LoRA α per layer as $\alpha^{(l)} = 2r^{(l)}$, so the effective update scale $\alpha/r = 2$ is held constant regardless of the rank assigned to a layer; the accuracy gaps we report are therefore not an artifact of varying per-layer effective learning rates. **GRPO:** $K = 4$ generations per prompt, $\beta = 0.05$ KL coefficient, 1000 training steps, learning rate 10^{-5} , single seed. We run rollout generation in-process with vLLM (*colocate* mode: the policy used for sampling shares GPU memory with the trainer, avoiding network transfer). **Eval:** 200 held-out GSM8K test problems; greedy decoding; Wilson 95% intervals.

3.2 Gradient Landscape Under GRPO

Figure 1 shows the reward sensitivity map under uniform LoRA. We note four properties:

- **Flat distribution.** Max/min importance ratio is $2.17\times$ (Layer 15 hottest at 4.68%; Layer 26 coldest at 2.15%). Even the coldest layer carries 46% of the hottest layer’s gradient signal.

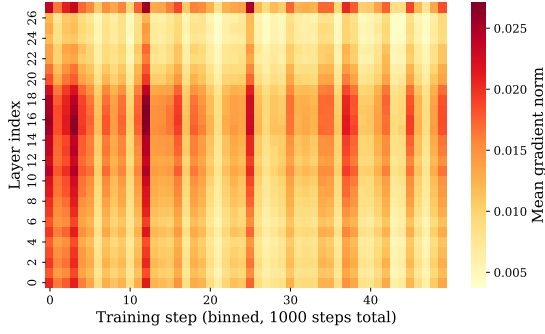


Figure 1: Per-layer adapter gradient magnitude during GRPO training (1000 steps, 28 layers, uniform $r = 32$). The distribution is notably flat relative to SFT-era reports.

Strategy	Params	Acc. (%)	95% CI
Base (no LoRA)	0	66.0	± 6.6
Uniform ($r=32$)	36.9M	74.5	± 6.0
Proportional ($r=20-40$)	36.9M	70.0	± 6.4
Random ($r=16-48$)	36.9M	67.5	± 6.5
Reduced 70% ($r=12-31$)	25.8M	65.0	± 6.6

Table 1: GSM8K accuracy under different rank allocations ($n=200$ test samples; single seed). All “same budget” methods use total rank 896 (28×32). Confidence intervals are Wilson score intervals.

- **Middle-layer concentration.** Layers 9–18 carry 43.0% of the gradient, but early (29.8%) and late (27.2%) layers also remain meaningful—unlike the SFT picture, where ILA (Shi et al., 2024) reports $>80\%$ concentration in the top 30% of layers.
- **Temporal stability.** The correlation between the per-layer importance profile in the first 250 steps and the last 250 steps is 0.962, indicating stable structural patterns rather than transient noise.
- **Module composition.** Attention modules (q/k/v/o) account for 52.2% of the gradient signal, FFN modules (up/down/gate) for 47.8%. The up_proj module is the single most reward-sensitive (21.7%; per-module breakdown in Appendix A).

3.3 Rank Allocation Results

Table 1 shows our main result: **uniform allocation outperforms every non-uniform variant we tested**, including gradient-aware proportional allocation, at the same parameter budget. Proportional allocation scores 4.5 points below

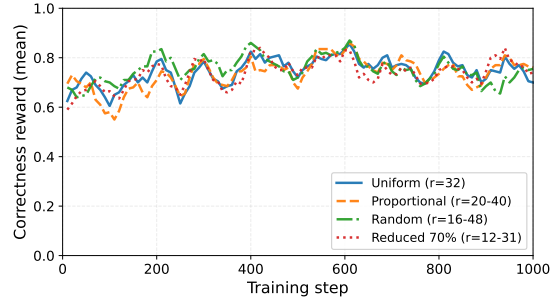


Figure 2: Training reward curves are nearly indistinguishable across all four configurations (Uniform, Proportional, Random, Reduced 70%); the evaluation gap in Table 1 emerges only on the held-out set.

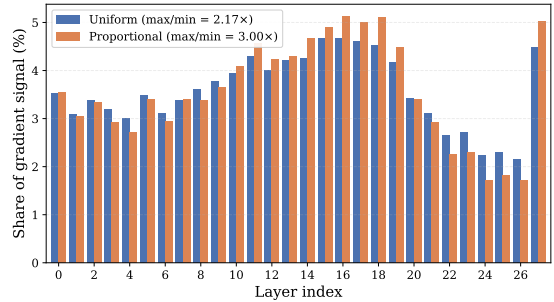


Figure 3: Normalized layer importance under uniform vs. proportional allocation. Non-uniform allocation amplifies the original importance spread from $2.17\times$ to $3.00\times$.

uniform. Random allocation scores even lower (67.5%). Reduced-budget allocation falls below the untrained base model. The confidence intervals overlap, so we treat the magnitudes as suggestive rather than statistically decisive (see Limitations).

Two observations stand out. First, **gradient-aware proportional allocation outperforms random allocation** (70.0% vs. 67.5%), so the importance signal is not noise—it identifies layers that matter. Yet both lose to uniform, so the signal, while directionally correct, is not actionable as a rank allocator under GRPO. Second, training reward curves are nearly identical across all four configurations (Fig. 2): correctness reward reaches 0.74–0.77 for every method by step 1000. The performance gap appears *only at evaluation*, suggesting that rank reallocation harms generalization rather than training dynamics.

3.4 The Gradient Amplification Effect

We profile gradients during *all* training runs, not just the uniform baseline. Figure 3 reveals a striking effect: the gradient importance spread *widens* under non-uniform allocation.

- Uniform: max/min ratio $2.17\times$.
- Proportional (same budget): $3.00\times$ (+38%).
- Reduced 70%: $3.57\times$ (+64%).

Layers given higher rank (e.g. Layer 18: $r = 40$) see their gradient share *increase* from 4.53% to 5.12%. Conversely, layers given lower rank (e.g. Layer 24: $r = 20$) see their share *decrease* from 2.23% to 1.71%. The allocation creates a positive feedback loop: more capacity $\rightarrow$ more gradient $\rightarrow$ appears even more “important”.

Critically, this effect is **causal, not correlative**. The random allocation experiment—where ranks are deliberately decoupled from gradient importance—shows the same amplification. The correlation between assigned rank and resulting gradient share is 0.972 for random and 0.946 for proportional. Under random allocation, Layer 1 (normally 3.09% of gradient) receives $r = 48$ and jumps to 4.21%; Layer 10 (normally 3.93%) receives $r = 16$ and drops to 2.85%. **Rank determines gradient importance, not the other way around.**

This has a methodological consequence: gradient profiling under one allocation cannot reliably inform a different allocation. The “profile then re-train” paradigm common in SFT-era adaptive-rank methods is unstable for GRPO, at least at this scale. One natural concern is whether the effect is trivially explained by more parameters yielding larger aggregate gradient norms. Equation 2 aggregates gradient norms within a layer, so larger layers can mechanically produce larger $S(l)$. However, the same amplification persists when restricted to individual modules at identical dimensions (e.g. q_proj at 1536×1536 across all layers; Appendix A), so it is not purely a parameter-counting artifact.

3.5 Why SFT Methods May Fail Under RL

We sketch an interpretation. Under SFT the loss is per-token cross-entropy and a small number of layers dominate the gradient landscape (Shi et al., 2024: >80% in the top 30%), so there are genuinely idle layers whose capacity can be redistributed. Under GRPO, the loss is an advantage-weighted policy gradient with sparse, binary reward; the top 30% of layers in our profile carry only 35.7% of the signal, and even the coldest layers contribute 2.15%. Reducing their capacity—even modestly, from $r = 32$ to $r = 20$ —damages evaluation accuracy while leaving training reward

unchanged. We conjecture that “cold” layers in this setting handle structural functions (output formatting, numerical precision, coherence) that are already well-handled by the pretrained model and become bottlenecks only when capacity is reduced. The near-even attention/FFN split (52.2%/47.8%) is consistent with this story, unlike SFT where FFN layers in the upper half dominate (Zhang et al., 2023).

4 Related Work

Adaptive rank for SFT. AdaLoRA (Zhang et al., 2023) prunes singular values during training. GoRA (He et al., 2025) uses gradient-weight products for initialization-time allocation. IGU-LoRA (Cui et al., 2026) applies integrated gradients with uncertainty-aware scoring. Aletheia (Saket, 2026) selects layers via gradient probes. All of these operate exclusively under supervised objectives.

Layer importance in alignment. ILA (Shi et al., 2024) learns binary layer masks and shows that 10–30% of layers suffice for SFT alignment. Young (2026) argue that RLHF gradients concentrate at specific positions. Our work extends this line by showing that even when RL gradients are non-uniform, their concentration is too mild to support useful rank reallocation.

GRPO and LoRA. DeepSeekMath (Shao et al., 2024) introduced GRPO for mathematical reasoning. To our knowledge, no prior work has investigated adaptive rank allocation specifically under RL-based alignment.

5 Limitations

The result is *preliminary* and the following caveats matter. The experiments use one model (Qwen 2.5 1.5B), one dataset (GSM8K), one RL algorithm (GRPO), and single-seed runs with $n = 200$ test problems. The Wilson intervals in Table 1 overlap, so the 4.5-point gap is suggestive rather than statistically decisive. We compare to SFT gradient distributions only via prior work (Shi et al., 2024; Zhang et al., 2023); a matched SFT baseline on the same model and data would sharpen the contrast. We hold LoRA scaling constant across allocations ($\alpha/r = 2$ per layer), but have not isolated the contribution of adapter initialization or optimizer state from the rank itself. Finally, our cold-layer hypothesis is untested; per-reward gradient decomposition (correctness vs. format) could partially verify it. We encourage replication at larger scale.

6 Lessons and Conclusion

In our setting, the gradient-based rank allocation that works under SFT does not transfer to RL alignment via GRPO: proportional allocation degrades accuracy by 4.5 points at an identical parameter budget, random allocation degrades further, and a random control shows that rank itself causally shapes the resulting gradient distribution. The mechanism—a flat gradient landscape combined with a positive-feedback amplification effect—suggests three practitioner-facing lessons. (i) Gradient importance is a useful diagnostic but not an allocator under GRPO: the signal ranks layers correctly (proportional beats random) but does not justify deviating from uniform at a fixed budget. (ii) Profile-then-retrain is unstable under RL, because the profile changes with the allocation ($r = 0.97$ between assigned rank and post-hoc importance). (iii) Methods that exploit concentrated gradient structure under SFT cannot be assumed to work under RL, where the structure we observe is much flatter. Future work should test whether dynamic, training-time rank adaptation can break the feedback loop, and whether the effect persists at larger scale, on other tasks, and under other RL objectives such as PPO and DPO.

References

- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Xuan Cui, Huiyue Li, Run Zeng, Yunfei Zhao, Jinrui Qian, Wei Duan, Bo Liu, and Zhanpeng Zhou. 2026. IGU-LoRA: Adaptive rank allocation via integrated gradients and uncertainty-aware scoring. *arXiv preprint arXiv:2603.13792*.
- Haonan He, Peng Ye, Yuchen Ren, Yuan Yuan, Luyang Zhou, Shucun Ju, and Lei Chen. 2025. GoRA: Gradient-driven adaptive low rank adaptation. In *Advances in Neural Information Processing Systems*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Abdulmalek Saket. 2026. Aletheia: Gradient-guided layer selection for efficient LoRA fine-tuning across architectures. *arXiv preprint arXiv:2604.15351*.

Module	Family	Share (%)	Top-3 layers
up_proj	FFN	21.7	L27, L3, L18
gate_proj	FFN	17.1	L27, L18, L17
v_proj	Attn	15.6	L0, L11, L15
o_proj	Attn	14.9	L0, L15, L16
k_proj	Attn	12.4	L18, L17, L19
q_proj	Attn	9.3	L16, L14, L9
down_proj	FFN	8.9	L1, L27, L2
Attention total		52.2	—
FFN total		47.8	—

Table 2: Per-module share of total adapter gradient magnitude under uniform $r = 32$ GRPO training on GSM8K (mean over final 200 steps; B matrices). The attention/FFN split is near-even (52.2%/47.8%), and up_proj is the single most reward-sensitive module.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangyuan Shi, Zexin Lu, Xiaoyu Dong, Wenlong Zhang, Xuanyu Zhang, Yujie Feng, and Xiao-Ming Wu. 2024. Understanding layer significance in LLM alignment. *arXiv preprint arXiv:2410.17875*.

Robin Young. 2026. Why is RLHF alignment shallow? A gradient analysis. *arXiv preprint arXiv:2603.04851*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*.

A Per-Module Gradient Importance

Table 2 reports the average per-module gradient magnitude across all 28 layers under the uniform $r = 32$ baseline. Values are aggregated from the B adapter matrix at each layer (the A matrix shows the same qualitative ordering at smaller magnitude) and averaged over the final 200 steps of GRPO training. The split between attention (q/k/v/o) and FFN (up/down/gate) modules is near-even (52.2% vs. 47.8%), with up_proj the single most reward-sensitive module.

Do Not Score the Edited Answer: A Negative Result on Loose Evaluation for Verifiable Instruction Following

José Luis Delgado

Universitat Oberta de Catalunya

Barcelona, Spain

jdelgado13@uoc.edu

Abstract

Verifiable instruction-following benchmarks often include strict and loose scoring modes: strict scoring checks the original response, and loose scoring applies the checker to transformed versions of the response, for example after removing markdown or dropping boundary lines, to recover false negatives caused by harmless wrappers and formatting artifacts. We study a negative result for this repair strategy using a constructive suite of 144 canaries in which every valid wrapper is paired with an invalid deletion trap that has the same transformed-response checker outcomes under IFEval-style transformations. Loose scoring accepts all valid wrappers and all invalid traps; support over transformation functions accepts both classes, and a threshold that deduplicates passing edited strings rejects both classes. On the balanced suite, any scorer that observes only the transformed-view pass signature has a best possible accuracy of 0.50, and the canaries are solved only by a boundary-aware certificate that inspects the original deleted span. Deletion- or normalization-based loose evaluation therefore needs evidence-preservation checks when edited spans may affect instruction satisfaction.

1 Introduction

Verifiable instruction-following evaluation uses deterministic checkers to replace open-ended judgment with executable constraints. IFEval is the canonical example: each prompt contains one or more verifiable instructions, and each instruction is scored by a corresponding verification function (Zhou et al., 2023). FollowBench and IFBench extend the same line of work through richer constraints and larger instruction inventories (Jiang et al., 2024; Pyatkin et al., 2025). These benchmarks reduce dependence on preference modeling, although their scores still depend on the response object supplied to the checker.

The strict and loose modes used by IFEval expose this design choice directly. Strict scoring applies a checker to the original model response, whereas loose scoring applies the checker to several transformed response views, such as versions with markdown removed or the first or last line dropped, and accepts the response when any transformed view passes. Loose scoring can repair cases where an otherwise correct answer is surrounded by boilerplate or formatting text that causes a brittle exact checker to fail.

The same strategy can also delete evidence that determines whether the original response satisfies the instruction. If the removed line is a harmless wrapper, loose scoring repairs a false negative; if the removed line violates the instruction, loose scoring evaluates an edited answer that no longer contains the violation. The difference is semantic, but the loose criterion only observes checker outcomes on transformed strings.

Zhou et al. note that loose scoring can introduce false positives (Zhou et al., 2023). The canaries formalize that observation as an indistinguishability condition: for deletion-based loose scoring, a valid repair and an invalid deletion trap can have the same transformed-view pass signature, so no rule based only on that signature can separate them.

The self-contained canary suite specifies an answer line and a boundary policy. Boundary lines may appear before, after, or around the answer line, but they must not violate a stated policy such as containing a forbidden word, comma, or digit. For each setting, we construct a valid wrapper and an invalid deletion trap with the same target answer and the same transformed-view checker outcomes. The valid case contains a harmless boundary line, and the invalid case contains a boundary line that violates the policy. After the relevant boundary line is dropped, both reduce to the same answer line.

The suite isolates transformed-response scoring from model behavior: the exact-checker strict base-

line rejects all canaries because the original response includes boundary text, while loose scoring accepts all valid wrappers and all invalid deletion traps. A support-count filter that requires at least two passing transformed views accepts the same cases, because the valid and invalid members of each pair have identical passing-view signatures. More generally, any classifier whose evidence is limited to the transformed-view pass signature, the target, the boundary position, and the instruction policy has a best possible in-sample accuracy of 0.50 on the balanced suite.

Comparing signature-only rules with a simple boundary certificate identifies the missing evidence. The certificate accepts a transformed-view pass only when the deleted boundary span does not violate the original policy, and it solves the suite by using the original response span removed by the transformation. As a diagnostic for this canary family, the certificate shows that loose scoring by edited responses requires an evidence-preservation check when the edit can delete instruction-relevant text.

2 Loose Scoring by Response Transformation

Let I be an instruction and y a model response. The predicate $V_I(y)$ denotes validity of the original response under the full instruction, including any boundary or formatting condition, while $e_I(x)$ denotes a deterministic verification function applied to a particular string x . The checker may be brittle with respect to wrapper text. Strict scoring applies the checker to the original response,

$$z_{\text{strict}}(I, y) = e_I(y),$$

whereas loose scoring accepts if any transformed response passes,

$$z_{\text{loose}}(I, y) = \max_{t \in T} e_I(t(y)),$$

where T is a set of response transformations. This form covers the standard loose-evaluation pattern in which wrappers or formatting artifacts are removed before checking. The canary result concerns scorers that approximate $V_I(y)$ using only checker outcomes $e_I(t(y))$ over transformed response views.

For a fixed I , define the transformed-view pass signature

$$\sigma_I(y) = (e_I(t(y)))_{t \in T}.$$

The support count is $\sum_{t \in T} e_I(t(y))$. A support-count filter is still a function of $\sigma_I(y)$; it can require more than one passing view, but it cannot inspect content that all relevant passing transformations removed.

Indistinguishability. Suppose two responses y^+ and y^- satisfy $V_I(y^+) = 1$, $V_I(y^-) = 0$, and $\sigma_I(y^+) = \sigma_I(y^-)$. Any scoring rule $g(I, \sigma_I(y))$ assigns the same verdict to both responses and must be wrong on at least one of them. If a canary suite is balanced into valid and invalid pairs with identical signatures, the best possible accuracy for any such rule is 0.50. The bound follows from the information available to the scorer and does not depend on model behavior.

3 Constructive Canaries

The suite contains 144 canaries obtained by crossing six target answers, three boundary positions, and four boundary policies, with one valid wrapper and one invalid deletion trap for each setting. The target answers are blue, Yes, done, cat, Paris, and 42. Boundary lines may appear as a prefix, suffix, or both, and the policies forbid one of four surface properties: the word red, the word banana, a comma, or a digit.

Each instruction has the same form: the answer line should be exactly the target, and boundary lines may appear in the specified position but must not violate the stated policy. The checker is brittle by construction: it passes only when the checked string is exactly the target answer. The transformation set contains the eight IFEval-style transformations used in the experiment: identity, mark-down removal, first-line deletion, last-line deletion, first-and-last-line deletion, and the corresponding markdown-removal compositions. Because paired cases differ only in whether the deleted boundary line violates the policy, the suite isolates the scoring issue from response generation.

The canaries are synthetic, but their ingredients match ordinary verifiable-instruction components: exact constrained answers, forbidden tokens, punctuation constraints, and beginning, ending, or whole-response conditions. IFEval-style loose transformations explicitly remove first or last lines to skip boilerplate, which is the operation used by the deletion traps.

In the representative pair in Table 1, the valid wrapper is acceptable under the original instruction because answer note does not violate the red

Case	Original response	After deletion
Valid wrapper	answer note	
blue	blue	
Deletion trap	red note	
blue	blue	

Table 1: Representative canary pair for an instruction that requires the answer line to be exactly blue and allows a prefix line only if it does not contain red; dropping the first line makes both responses identical to the checker.

Method	Valid recall	Invalid accept	Acc.
Exact strict	0.00	0.00	0.50
Loose edited-view	1.00	1.00	0.50
Support ≥ 2	1.00	1.00	0.50
Unique support ≥ 2	0.00	0.00	0.50
Boundary certificate	1.00	0.00	1.00

Table 2: Synthetic canary results for exact strict scoring, loose edited-view scoring, support thresholds, and the boundary certificate; only the boundary certificate inspects the original deleted boundary span.

policy, while the deletion trap is invalid because red note violates the policy. A first-line deletion transformation maps both responses to blue, so the transformed response contains no evidence about the boundary violation.

4 Results

The main canary results in Table 2 show that, under the brittle exact-answer checker, strict scoring has zero valid recall because every original response contains an allowed or disallowed boundary line and the checker only accepts the bare answer. Loose edited-view scoring reaches perfect recall on valid wrappers and accepts every invalid deletion trap for the same reason. Requiring at least two passing transformation functions does not change the outcome, since the paired valid and invalid cases have the same passing transformations.

The unique-support row deduplicates passing transformed strings before counting support. It is not a content-aware scorer: it ignores the contents of failing views, including the original response. Under this rule, each canary has one unique passing string, the bare target, so a threshold of two rejects both valid wrappers and invalid traps, yielding the same 0.50 accuracy by trading false positives for false negatives.

We report the unique-support variant as a support-threshold heuristic, separate from the

Quantity	Value	Share
Signature groups	72	–
Canary cases	144	1.00
Minimum signature errors	72	0.50
Upper-bound accuracy	–	0.50

Table 3: Best possible in-sample accuracy for any classifier that observes only the transformed-view pass signature, target, deleted-span position, and instruction policy; mixed-signature groups force at least one error per valid/invalid pair.

Deleted span	Policy	Valid	Invalid
both	digit	6	6
prefix	comma	6	6
prefix	red	6	6
suffix	banana	6	6
All signatures	all	72	72

Table 4: Examples of indistinguishable transformed-view signatures; each row aggregates six target-specific mixed groups in which the valid wrapper and invalid deletion trap share the same target, deleted-span position, policy, and passing transformed views.

signature-only upper bound in Table 3. A content-aware scorer that inspects the original response is outside that evidence restriction and corresponds to the boundary certificate.

In the boundary-certificate row, the scorer solves the suite by inspecting the original deleted span. The certificate first requires a transformed-view pass, identifies the boundary span deleted by the passing transformation, checks that span against the original policy, and accepts only if the deleted span does not violate the policy. It accepts every valid wrapper and rejects every invalid trap, which identifies whether the transformation removed instruction-relevant evidence as the missing information.

The corresponding indistinguishability bound in Table 3 follows from 72 mixed signature groups, each containing one valid wrapper and one invalid deletion trap with the same target, deleted-span position, instruction policy, and transformed-view pass signature. Any classifier restricted to that evidence assigns the same label to both members of each pair, producing at least 72 errors on the balanced suite.

The grouped signatures in Table 4 aggregate over target answers within a fixed deleted-span position and policy. The total row covers all mixed groups, where the view-signature evidence is exactly bal-

anced between valid and invalid cases. Additional passing views therefore do not constitute semantic confirmation; passing views only say that some edited response satisfies the checker.

5 Discussion

The negative result applies to loose evaluation schemes that score edited responses without certifying that the edit preserves evidence needed by the original instruction, including support-count variants that coarsen the transformed-view signature. The construction does not require model-specific adversarial behavior; it applies whenever a model output contains instruction-relevant boundary text that a loose transformation deletes. We do not estimate how often such traps occur in deployed outputs, since the result is an information-loss canary with no prevalence estimate.

For response transformations used to repair brittle checkers, the boundary certificate gives the relevant evidence-preservation condition: the transformation should be paired with a certificate over the original response span. For boundary deletion, that certificate must check whether the removed span is irrelevant to the instruction being scored. For formatting normalization, it must check whether the normalized surface preserves the property under evaluation. Without that evidence-preservation check, loose scoring measures an edited answer and not the model response that was actually produced.

Scope beyond deletion. The indistinguishability argument is not specific to line deletion. It applies to any many-to-one transformation for which a compliant output and a violating output map to the same transformed view and therefore induce the same checker signature. Markdown stripping, case folding, whitespace normalization, and punctuation removal inherit the same impossibility whenever the erased distinction is instruction-relevant. Conversely, an injective normalization over the relevant output space does not create this particular information-loss failure. The risk is therefore tied to lossy transformations, rather than to normalization per se.

The canary construction aligns with prior work on evaluation sensitivity to formatting and surface variation (Sclar et al., 2024). It follows the logic of behavioral testing by isolating a failure mode of the evaluation procedure, with no estimate of failure rates for any particular model distribution (Ribeiro et al., 2020).

6 Conclusion

Loose scoring by response transformation is reliable as a repair only when the transformation preserves the evidence needed by the original instruction. The canary suite gives an exact failure case for deletion-based loose scoring: each valid wrapper and its invalid deletion trap have the same target answer and the same transformed-view pass signature after deletion, so any signature-only scorer is limited to 0.50 accuracy on the balanced suite. Counting more passing transformations does not add semantic evidence, and deduplicating passing edited strings changes the error type without recovering the deleted span.

By solving the suite only after reading the original span removed by the transformation, the boundary certificate identifies the missing variable. For verifiable instruction following, a loose score obtained from an edited response should be interpreted as a score for that edited response unless the scorer also certifies that the edit preserved instruction-relevant evidence. The canaries turn this requirement into a testable condition for deletion- and normalization-based repairs.

Limitations

The canaries are synthetic and balanced by construction, so they establish an information-loss failure mode without estimating its empirical frequency for deployed model outputs. The suite tests exact-answer constraints with boundary policies over prefix and suffix text; other instruction families used in verifiable instruction-following benchmarks would require additional canary constructions. The boundary certificate is included as a diagnostic for these canaries, and a production evaluator for broader constraint families would require instruction-specific evidence-preservation rules.

Our balanced synthetic canaries establish possibility and an information-theoretic limitation; they do not estimate the prevalence of deletion traps in naturally generated model outputs. Estimating that prevalence would require a separately sampled corpus of raw outputs and human adjudication of whether discarded boundary material violates the original instruction. Accordingly, the 0.50 bound should not be interpreted as an estimate of the benchmark-wide false-positive rate. It characterizes the worst-case identifiability of evaluators restricted to transformed-view signatures.

Ethical Considerations

The study uses synthetic text generated by deterministic scripts and does not involve human subjects, private data, or model-generated personal information. The main risk is misuse of the result as a claim that loose scoring is always invalid.

References

- Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. [Follow-Bench: A multi-level fine-grained constraints following benchmark for large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688, Bangkok, Thailand. Association for Computational Linguistics.
- Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. 2025. [Generalizing verifiable instruction following](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. [Beyond accuracy: Behavioral testing of NLP models with CheckList](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.
- Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. [Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting](#). In *The Twelfth International Conference on Learning Representations*.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *Preprint*, arXiv:2311.07911.

Perturbation, Not Provenance: Why Activation Steering Does Not Make a Robust Text Watermark

Sergey Pletenev^{1,2}

¹AIRI, ²Skoltech

alex010rey@gmail.com

Abstract

Linear directions in the activation space of large language models causally and universally influence model behavior. This suggests that they might serve as a basis for a robust text watermark. The idea is to steer the generation along a specific semantic direction and then recover the mark by probing the activations of the re-encoded text. We test this hypothesis by conducting a series of experiments. We vary the strength of the steering and use four different directions (formality, sentiment, complexity, and a random control) across two different model families. We also test two paraphrasing attacks: a sentence-level one and a stronger paragraph-level one. The results show that activation-steering watermarks are outperformed by a standard logit-based watermark (KGW) in terms of detection accuracy. At a matched level of fluency (within 1.15 times the original text’s perplexity), KGW achieved an AUC of 0.81 against a sentence-level paraphraser, while the best steering direction only reached 0.65. A stronger paragraph-level paraphraser lowered both (0.65 for KGW and 0.59 for steering), narrowing the gap but not reversing it. On Qwen, it left the two indistinguishable but at near-chance levels for both. We further showed that the detectability provided by steering was not semantic: a random direction was just as detectable as a meaningful one. The signal was independent of which layer was probed and whether a surface feature control was used. We conclude that, in this case, semantic activation does not encode a localized, recoverable semantic mark, but rather a global, content-independent shift that can only be strongly detected when the text is degraded.

1 Introduction

Text watermarking is a process of embedding a detectable signal into machine-generated text, allowing a third party with a specific key to later verify the text’s authenticity. A key aspect of watermarking is its resistance to paraphrasing, meaning that

an adversary attempting to rewrite the text should not be able to easily remove the watermark. Watermarks based on logit functions, such as KGW (Kirchenbauer et al., 2023), bias token sampling towards a pseudo-random “green list”. On the other hand, there are semantic watermarks, like SemStamp (Hou et al., 2024a), which target paraphrasing resistance by operating in the space of sentence embeddings.

A natural and previously unexplored alternative is to embed a watermark inside the forward pass of the model. Recent work has shown that linear directions in the residual stream can causally steer the behavior of the model and generalize across different contexts. (Turner et al., 2023; Rimskey et al., 2024; Zou et al., 2023; Park et al., 2024). If a semantic direction leaves a trace that survives paraphrasing because new sentences preserve meaning even when they rewrite tokens, then we could: (i) steer generation along a secret semantic direction and (ii) detect the watermark by re-encoding the candidate text through the base model and probing its activations. Such a watermark would be architecturally different from logit- and embedding-space methods and might inherit the robustness of the underlying semantic representation to paraphrasing.

We tested this hypothesis and reported a negative result. Our main contributions are:

1. We formalize and implement an activation-steering watermark and evaluate it against KGW on a detection-versus-fluency frontier, under both a sentence-level and a paragraph-level paraphrase attack, across steering strengths spanning fluent to degraded generation.
2. We show that the steering watermark is Pareto-dominated: at matched deployable fluency, it is never more detectable than KGW, and only reaches comparable detectability by inflating the perplexity by several-fold.

3. We demonstrate that its detectability is not semantic via three controls: a random steering direction matches meaningful ones, the signal is layer-invariant, and it cannot be explained by surface text statistics.

We release data and code for further usage¹.

2 Related Work

Logit-based watermarks. KGW (Kirchenbauer et al., 2023) and successors (Zhao et al., 2024) partition the vocabulary and bias sampling; detection is a statistical test over token counts. These methods are known to degrade under paraphrase, motivating semantic approaches.

Semantic / paraphrase-robust watermarks. SemStamp (Hou et al., 2024a) and its variants (Hou et al., 2024b) operate in sentence-embedding space via rejection sampling to survive paraphrase. Our method differs by intervening in the residual stream; we find this confers no robustness advantage over the logit baseline at matched fluency.

Activation steering. Linear directions in activation space causally and generalizably affect model behavior (Turner et al., 2023; Rinsky et al., 2024; Zou et al., 2023; Park et al., 2024), which motivated the watermarking hypothesis we test. Our negative result delimits that promise: such directions do not yield a localized, recoverable semantic channel that survives paraphrase.

Concurrent work, SLAM (Harel-Canada and Sahai, 2026), instantiates an activation-injection watermark using sparse-autoencoder structural directions (voice, tense, clause order) on Gemma-2. Our results delimit this specific case – the directions that transmit are behavioral and therefore collide with natural text. Steered structure that is not lexically anchored is what a paraphrasing tool removes. We test a sentence-level paraphrasing tool only, so we do not claim to overcome structural marks in SLAM; rather, we describe the secrecy-robustness tension that any such system must navigate.

3 Method

Watermark by activation steering. Let $h_t^{(\ell)} \in \mathbb{R}^d$ be the residual-stream activation at layer ℓ and token position t . Given a unit-norm steering direction $v \in \mathbb{R}^d$, we add, at every position during

generation,

$$\tilde{h}_t^{(\ell)} = h_t^{(\ell)} + \alpha \|h_t^{(\ell)}\| v, \quad (1)$$

where α is a *norm-relative* steering fraction. Scaling by $\|h_t^{(\ell)}\|$ makes α comparable across models and layers whose residual norms differ by orders of magnitude (e.g. $\approx 10^2$ for Llama-3.2-1B at $\ell=8$ versus $\approx 2.2 \times 10^3$ for Qwen3-1.7B at $\ell=14$); an absolute offset that steers one model is negligible or destructive for another.

Detection. The verifier re-encodes candidate text through the unmodified base model, mean-pools the layer- ℓ activations over non-padding positions, and trains a regularized logistic probe to separate steered from unsteered text. We report detection as the cross-validated ROC-AUC of this probe (pooled across generation seeds).

Directions. We extract steering directions by mean-difference of pooled activations over contrastive text pairs (Rinsky et al., 2024) generated by a larger model of the same family: formality (formal vs. informal), sentiment (positive vs. negative), and complexity (complex vs. simple). As a control we use a *random* unit direction orthogonalized against formality. If steering encodes a semantic mark, the meaningful directions should be detectable and the random one should not.

Baselines. We compare against KGW (Kirchenbauer et al., 2023) as implemented in, sweeping its logit bias δ . KGW is detected with its native z -score test, not the activation probe. For paraphrasing we use two attackers. The first rewrites every generation sentence-by-sentence with a Pegasus (Zhang et al., 2020). Because a sentence-level paraphraser is not able to reorder or merge material across sentence boundaries, we also use DIPPER (Krishna et al., 2023), a 11B paragraph-level paraphraser that rewrites in few sentence windows.

4 Experimental Setup

Models. We use Llama-3.2-1B-Instruct (Grattafiori et al., 2024) and Qwen3-1.7B (Qwen Team, 2025) as generators, with the 8B model of each family used only to synthesize contrastive pairs.

Prompts and generation. We sample 200 Alpaca instruction prompts and generate with temperature 1.0, top- p 0.9 and 150 max new tokens. Every operating point is generated under 3 seeds (600 generations per class).

¹<https://github.com/AlexRey/Steering-Text-Watermarking>

Methods	Ppl.	d	AUC _{Peg}	AUC _{DIP}
<i>Deployable regime</i> ($\leq 1.15 \times \text{ppl}$)				
KGW ($\delta=3$)	1.06	0.06	0.81	0.65
sentiment ($\alpha=0.1$)	1.03	0.03	0.65	0.59
random ($\alpha=0.2$)	1.13	0.12	0.61	0.57
complexity ($\alpha=0.4$)	1.03	0.03	0.59	0.55
formality ($\alpha=0.2$)	1.05	0.05	0.57	0.52
<i>Steering's high-detectability points</i>				
formality ($\alpha=0.6$)	1.73	0.55	0.91	0.83
sentiment ($\alpha=0.2$)	1.32	0.28	0.89	0.83
random ($\alpha=0.4$)	1.98	0.68	0.86	0.78

Table 1: Llama-3.2-1B, best point per method within each case, under the Pegasus (Peg) and DIPPER (DIP) attacks. Ppl. is the ratio to unsteered and $d = |\log(\text{ppl ratio})|$

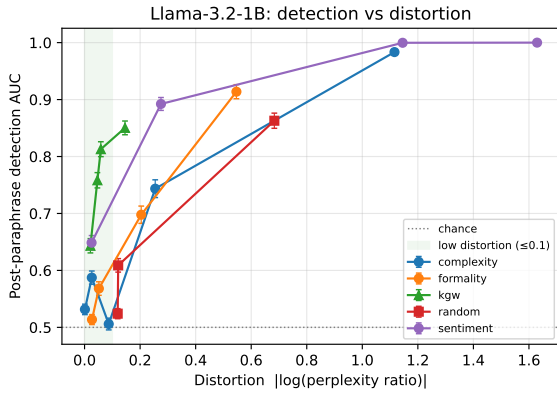


Figure 1: Detection-versus-distortion frontier (Llama-3.2-1B)

Steering grid. The norm-relative fraction α is swept per model over a grid chosen by calibration to span fluent-but-changed to degraded generation ($\alpha \in \{0.1, 0.2, 0.4, 0.6, 0.8\}$). Because all figures plot detection against *measured* perplexity rather than against α , per-model grids remain directly comparable in fluency space.

Metrics. In both frontier figures (Figures 1 and 2) and in Table 1, the y -axis is post-paraphrase detection AUC; we report symmetric AUC ($\max(\text{AUC}, 1 - \text{AUC})$) so a key-aware verifier’s separability is never under-counted. For the x -axis we use a *two-sided distortion* $d = |\log(\text{ppl}_{\text{steered}}/\text{ppl}_{\text{unsteered}})|$, the absolute log of the perplexity ratio under the base model. A one-sided “fluency cost” (perplexity ratio) is not adequate here because, on Qwen, several steering directions lower perplexity relative to the unsteered baseline, we analyze this in Results section.

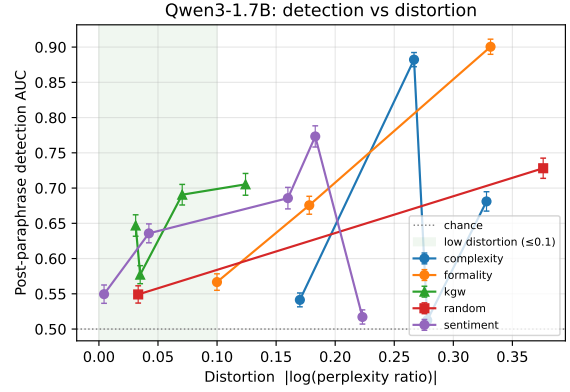


Figure 2: Detection-versus-distortion frontier (Qwen3-1.7B)

5 Results

Table 1 and Figures 1 and 2 show the main results. In the deployable case, KGW achieves 0.81 post-paraphrase AUC with a perplexity of 1.06. No other approach comes close to this level of performance at comparable fluency. Sentiment achieves the best result with 0.65 AUC, and the remaining three approaches fall between 0.57 and 0.61. Steering matches KGW’s best operating point (0.85) only from a perplexity of 1.16 upward. Non-sentiment approaches only perform well at $1.7\text{--}2\times$, which corresponds to a noticeable degradation in text quality. The logit watermark dominates the activation-steering watermark across the frontier, except for one case: on Qwen, sentiment with $\alpha = 0.8$ reaches 0.77, which is higher than KGW’s best (0.71), but at a greater distortion ($d = 0.18$ compared to 0.12).

Because both detectors score the same unsteered negative set, their AUCs are correlated, so we bootstrap the *difference* directly, resampling whole prompts (each prompt contributes one generation per seed to each class) and recomputing both AUCs on every replicate. In the deployable band KGW exceeds the best steering direction by 0.16 AUC under Pegasus, and no replicate put the difference at or below zero. The curves do not overlap within their error bars anywhere in the low-distortion regime (Figure 1).

A stronger attacker narrows the gap. DIPPER is much more damaging to KGW than Pegasus. When the attack is turned on, KGW loses an average of 0.15 AUC on Llama and 0.11 AUC on Qwen over its four different bias settings, while the four $\alpha = 0.1$ points lose only 0.01 AUC, an

order of magnitude less on both. This is expected, as KGW is more susceptible to the attack than the other models.

A paragraph-level rewriter disrupts the token-level green list, whereas a probe on mean-pooled activations reads a global perturbation that partly survives rewording. On Llama, the gap therefore falls from 0.16 AUC under Pegasus to 0.06 under DIPPER, so Pegasus alone is an artifact of a weak attacker. On Qwen, the ordering does not survive: the best in-band steering point (sentiment, $\alpha=0.4$) reaches 0.587 against KGW’s 0.557, a difference of -0.03 with interval $[-0.07, +0.01]$. Under the stronger attack, the two are statistically indistinguishable on Qwen, and we do not claim KGW wins there. However, Qwen is the model on which KGW performs the worst, so the convergence reflects a decreasing baseline as well as an increasing steering curve. 0.587 is close enough to the chance level to be interpreted as neither method being effective here, rather than the steering curve succeeding.

At the top of the steering grid, generation degenerates into empty or repetitive text whose perplexity ratio exceeds $10\times$ the unsteered baseline. We exclude these points from all figures and tables: they are not deployable watermarks but destroyed text.

Steering can reduce perplexity On Qwen3-1.7B several steering directions produce a perplexity ratio below 1: steered generations are *more* probable under the base model than unsteered ones (formality and complexity at $\alpha \leq 0.4$ give 0.72–0.90). Steering towards "formal" or "complex" registers did not degrade fluency so much as homogenized output towards higher-probability, more generic phrasing.

This gives us a methodological implication: detectability rises even as perplexity falls, so it cannot be a function of "fluency cost" alone. What the detector tracked was displacement from the base distribution in either direction, which was why our figures used the two-sided axis d . Under it, the apparent "free detection" of these points resolved – they sat at moderate distortion ($d = 0.1$ – 0.33 , not at zero cost).

A second implication was that perplexity alone could not decide which points were deployable. Strong steering failed predominantly by *repetition*, and repeated tokens are highly predictable, so a collapsed completion has a *low* perplexity. On Qwen, this puts formality $\alpha=0.6$ at a $1.15\times$ ratio

with 0.98 AUC – apparently a deployable watermark beating KGW – when one completion in ten has actually collapsed into a repetition loop. We therefore exclude operating points where more than 5% of completions are empty or have fewer than half their word 4-grams unique. On Qwen, this removes formality $\alpha=0.6$ and complexity $\alpha=0.8$; on Llama, it removes nothing that the perplexity rule hadn’t already removed, so Table 1. Appendix A.2 gives the criterion, an independent cross-model check confirming the effect is not an artifact of self-scoring, and per-point collapse rates. Detectability is achieved through degeneration, rather than a semantic marker.

6 Ablation study

If the steering encoded a semantic watermark, three predictions would hold: a meaningful direction would be more detectable than a random one, the signal would be strongest at the steered layer, and detection would not reduce to surface text statistics. However, the first two predictions fail outright, and the third is not refuted, but it is ineffective in the context that is relevant. At deployable fluency, there is no signal to explain surface statistics, because there is barely any signal at all. Where detection appears, it appears together with distortion, and it is available to a random direction as readily as it is to a meaningful one.

Random matches semantic. At matched steering strength the random control is among the most detectable directions, not the least. At $\alpha=0.4$ it reaches 0.86 AUC on Llama and 0.97 on Qwen, above formality (0.70 and 0.90), above complexity (0.59 and 0.68), and on Qwen above sentiment (0.64) as well; only Llama sentiment exceeds it. A direction carrying no semantic content is therefore no harder to recover than one carrying a concept.

Direction identity does not even transfer between models. Sentiment is the most detectable direction on Llama at every steering strength and the *least* detectable on Qwen, though both were extracted by the same mean-difference procedure from the same contrastive template.

Layer invariance. The signal is flat across layers. We probe every other layer across the full depth of each model and compare detectability. Post-paraphrase AUC varies by at most 0.044 (Llama, 8 layers) and 0.054 (Qwen, 14 layers) across all probed layers, and the layer we actually steered

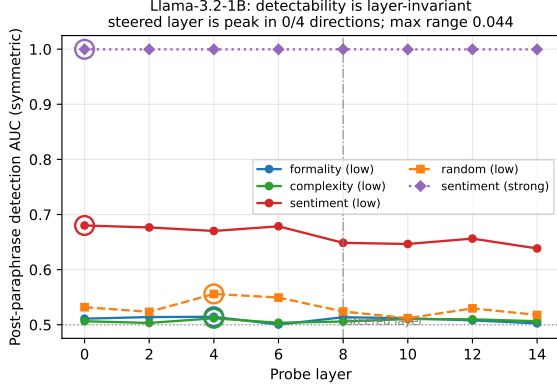


Figure 3: Post-paraphrase detectability against probe layer on Llama-3.2-1B. Rings mark each direction’s peak layer; the dash-dotted line is the layer that was steered. "low" is $\alpha=0.1$ and "strong" is $\alpha=0.4$.

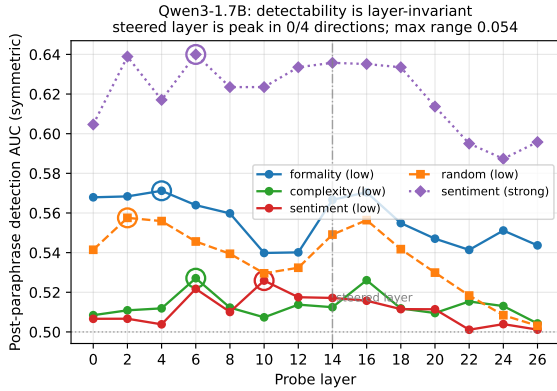


Figure 4: Post-paraphrase detectability against probe layer on Qwen3-1.7B

is the probe-AUC maximum in none of the four directions on either model (0/4 each); the peaks land at scattered, unrelated layers. A watermark localized to the steered representation would peak there. Instead the signal is a global property of the generated text, detectable at any depth (Figures 3 and 4). Because we cover the whole stack rather than a few layers, this is not an undersampling artifact, and because the pattern is the same in both model families it is not a configuration or anomaly of one architecture.

Surface statistics. At low, fluent steering ($\alpha=0.1$) neither probe separates the two sets. A probe trained only on shallow features (length, type–token ratio, punctuation rates) reaches 0.51–0.56 AUC, and the activation probe reaches 0.51–0.65; the two agree to within 0.02 for every direction on both models except sentiment on Llama, the single point where steering is detectable at deploy-

able fluency. Matching the two sets on completion length leaves the activation AUC essentially unchanged, so length is not the confound either. The correct reading is not that a deep signal survives a surface control, but that at fluent steering there is almost no signal to begin with.

At strong steering ($\alpha=0.4$) both probes rise together, and the surface probe becomes accurate exactly where the text is most distorted (0.90 for the random direction on Qwen, 0.81 for sentiment on Llama). Surface detectability therefore rises in lockstep with distortion for the random control just as for the meaningful directions, which is the same conclusion the other two controls reach: detection tracks displacement from the base distribution, not content.

The sentiment exception. Sentiment is the one direction that becomes strongly detectable at modest perplexity on Llama (AUC 0.89 at $1.32\times$). It is, however, also layer-invariant: heavy sentiment steering makes the entire text conspicuously more positive, a global shift detectable at any layer rather than a localized mark. Nor does it replicate—on Qwen the same direction is the weakest of the four at every steering strength, reaching only 0.77 even at $\alpha=0.8$. The exception is a property of one direction in one model, and it illustrates the mechanism rather than contradicting it.

7 Conclusion

Semantic activation-steering watermarks are an interesting idea, but they do not work as expected. In the testing regime we used, which included two model families with approximately 1-2 billion parameters, a single steered layer at mid-depth, and both sentence-level and paragraph-level paraphrasing attacks, a logit watermark always dominated the model, regardless of the fluency cost. What we found was that what steering provided was non-semantic: a random direction matched meaningful ones, the signal was layer-invariant, and what survived a surface-statistics control was content-independent.

Steering encoded a global perturbation that was only strongly detectable when it degraded text. We don’t claim that this rules out all activation-space watermarks – structural directions or larger models may behave differently – but it limits the semantic version of the idea, and we hope this saves others from taking a detour.

Limitations

We compare against a logit-based watermark (KGW) but not against a sentence-embedding watermark such as SemStamp (Hou et al., 2024a). That is arguably the more natural comparison: like ours, and unlike KGW, it is designed for paraphrase robustness by operating on meaning rather than on tokens.

We studied models with up to $\approx 2B$ parameters and steered a single mid-depth layer in each model. Larger models, where steering directions may be cleaner and more linear, could potentially change the picture. However, this is beyond the scope of this study, and it is possible that our setup tests a less favorable case for the steering hypothesis than a large-scale study would, so the negative result is unlikely to change with scale.

The concern about steering a single layer is also addressed by our design: the detector probes every layer and the signal is global rather than limited to the steered layer. Therefore, the claim does not depend on the specific layer being steered. Our detector is a deliberately strong key-aware probe; a more realistic key-blind detector would only lower steering detectability, reinforcing the negative result.

References

- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The Llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Fabrice Harel-Canada and Amit Sahai. 2026. [SLAM: Structural linguistic activation marking for language models](#). *Preprint*, arXiv:2605.05443.
- Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2024a. [SemStamp: A semantic watermark with paraphrastic robustness for text generation](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, NAACL 2024, Mexico City, Mexico, June 16-21, 2024, pages 4067–4082. Association for Computational Linguistics.
- Abe Bohan Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. 2024b. [k-SemStamp: A clustering-based semantic watermark for detection of machine-generated text](#). In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pages 1706–1715. Association for Computational Linguistics.
- John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. [A watermark for large language models](#). In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.
- Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. [Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. [The linear representation hypothesis and the geometry of large language models](#). In *Proceedings of the 41st International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 39643–39666. PMLR.
- Qwen Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Nina Rimskey, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. [Steering Llama 2 via contrastive activation addition](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 15504–15522. Association for Computational Linguistics.
- Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. [Steering language models with activation engineering](#). *Preprint*, arXiv:2308.10248.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. [PEGASUS: pre-training with extracted gap-sentences for abstractive summarization](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.
- Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. 2024. [Provable robust watermarking for AI-generated text](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xu Wang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2023. [Representation engineering: A top-down approach to AI transparency](#). *Preprint*, arXiv:2310.01405.

A Appendix

A.1 Full results

Tables 2 and 3 list every cases behind Figures 1 and 2, so the figures can be read numerically. They also make explicit something the figures cannot show: on Qwen the perplexity ratio is *not* monotone in the steering fraction, so the horizontal position of a point does not follow α . For complexity, $\alpha=0.1$ sits at $d=0.28$ while $\alpha=0.6$ sits at $d=0.27$; a line joining points in distortion order therefore visits the steering grid out of sequence. On Llama the ordering is preserved.

Method	α/δ	Ppl.	d	AUC _{Peg}	AUC _{DIP}
unsteered	–	1.00	–	0.50	0.50
formality	0.1	1.03	0.03	0.514	0.536
	0.2	1.05	0.05	0.568	0.519
	0.4	1.23	0.20	0.698	0.589
	0.6	1.73	0.55	0.914	0.831
	0.8	11.9	2.47	0.999 [†]	0.999 [†]
complexity	0.1	1.09	0.09	0.506	0.505
	0.2	1.00	0.00	0.532	0.513
	0.4	1.03	0.03	0.587	0.550
	0.6	1.29	0.25	0.744	0.644
	0.8	3.05	1.12	0.983	0.969
sentiment	0.1	1.03	0.03	0.649	0.593
	0.2	1.32	0.28	0.892	0.825
	0.4	3.14	1.15	1.000	0.998
	0.6	5.10	1.63	1.000	1.000
	0.8	12.0	2.48	1.000 [†]	1.000 [†]
random	0.1	1.13	0.12	0.524	0.508
	0.2	1.13	0.12	0.609	0.568
	0.4	1.98	0.68	0.863	0.780
	0.6	13.8	2.63	0.979 [†]	0.964 [†]
	0.8	86.9	4.46	0.996 [†]	0.990 [†]
KGW	1	1.02	0.02	0.643	0.558
	2	1.05	0.05	0.758	0.592
	3	1.06	0.06	0.813	0.652
	4	1.16	0.15	0.850	0.659

Table 2: Llama-3.2-1B, all operating points, under the Pegasus (Peg) and DIPPER (DIP) attacks. Bold marks the best KGW point inside the deployable band. [†] excluded from the frontier by the perplexity rule.

A.2 The degeneracy criterion

Why perplexity is insufficient. The frontier already discards points whose perplexity ratio ex-

Method	α/δ	Ppl.	d	AUC _{Peg}	AUC _{DIP}
unsteered	–	1.00	–	0.50	0.50
formality	0.1	0.90	0.10	0.567	0.534
	0.2	0.84	0.18	0.676	0.592
	0.4	0.72	0.33	0.900	0.772
	0.6	1.15	0.14	0.985 [†]	0.956 [†]
	0.8	–	–	0.987 [†]	0.977 [†]
complexity	0.1	0.76	0.28	0.512	0.521
	0.2	0.84	0.17	0.541	0.545
	0.4	0.72	0.33	0.681	0.588
	0.6	0.77	0.27	0.882	0.768
	0.8	2.37	0.86	0.980 [†]	0.956 [†]
sentiment	0.1	1.25	0.22	0.517	0.536
	0.2	1.00	0.00	0.550	0.550
	0.4	1.04	0.04	0.636	0.587
	0.6	1.17	0.16	0.686	0.677
	0.8	1.20	0.18	0.773	0.732
random	0.1	0.97	0.03	0.549	0.507
	0.2	1.46	0.38	0.728	0.670
	0.4	5834	8.67	0.967 [†]	0.930 [†]
	0.6	–	–	0.995 [†]	0.982 [†]
	0.8	3549	8.17	0.999 [†]	0.990 [†]
KGW	1	1.04	0.03	0.577	0.518
	2	1.03	0.03	0.647	0.532
	3	1.07	0.07	0.691	0.554
	4	1.13	0.12	0.705	0.557

Table 3: Qwen3-1.7B, all operating points, under the Pegasus (Peg) and DIPPER (DIP) attacks. Ratios below 1 are homogenization, not error. “–” marks generations too degenerate to score. Bold marks the best KGW point inside the deployable band. [†] excluded from the frontier by the perplexity rule.

ceeds $10\times$ as destroyed text. That rule cannot catch repetition, because a looping completion is highly predictable and therefore scores *low* perplexity. A point can be visibly destroyed and still sit in the deployable band. Qwen formality $\alpha=0.6$ is such a point: $1.15\times$ perplexity, 0.98 AUC, and completions such as “*The singular and singularly singular singular singular and singularly singular and singularly singular and...*”

This is not an artifact of self-scoring. Steering toward “formal” homogenizes text toward the steered model’s own high-probability phrasing, so one might suspect the ratio is depressed because the scorer is the model that was steered. It is not. Re-scoring every operating point under an independent Llama-3.1-8B judge places the same point at $1.13\times$, against $1.15\times$ self-scored. Both scorers agree, because the effect is a property of the text: no perplexity based axis can rank output whose dominant failure mode is one that perplexity rewards.

PEFT Gains Need Hidden Baselines: Rank and Optimizer Sweeps Change LoRA Conclusions

Ding Wu

Independent Researcher
dingwu@alumni.upenn.edu

Abstract

New parameter-efficient fine-tuning (PEFT) methods are often credited with gains against a single default LoRA configuration, even though inexpensive tuning of vanilla LoRA may change the comparison. We present a deliberately small, descriptive case study of two claims: adaptive rank allocation with AdaLoRA and LoRA-specific optimization with LoRA-RITE. Using RoBERTa-base on capped subsets of three GLUE classification tasks, we compare each method with a pre-specified vanilla LoRA control sweep. AdaLoRA beats fixed LoRA at rank 8 in 4 of 7 dataset–seed cells, but beats the best tested fixed rank in only 1 of 7. LoRA-RITE beats default AdamW in 3 of 6 cells, but tuned AdamW in only 1 of 6. Several SST-2 runs collapse to the majority label, so these counts are illustrative rather than population-level estimates. The practical contribution is an attribution protocol: report modest vanilla rank/learning-rate sweeps, per-cell margins, instability checks, and budget accounting before attributing gains to a specialized PEFT mechanism.

1 Introduction

Parameter-efficient fine-tuning adapts a pretrained model by training a small set of added or selected parameters while leaving most of the base model frozen (Houlsby et al., 2019; Hu et al., 2022). LoRA is especially common because it represents task updates with low-rank matrices and adds no inference-time module after merging (Hu et al., 2022). Recent methods improve this recipe through adaptive rank allocation, such as AdaLoRA (Zhang et al., 2023), or through optimizers specialized for LoRA’s factorization, such as LoRA-RITE (Yen et al., 2025).

These comparisons depend on what counts as the vanilla LoRA baseline. A method can look useful against one default rank and learning rate but less distinctive once vanilla LoRA receives a small,

method-agnostic sweep. We call this omitted control a *hidden baseline*: a pre-specified sweep that is cheap enough to report but is often absent from the headline comparison. It is neither a new method nor an oracle over an unlimited search space.

We audit this issue in a low-budget encoder-classification setting. GLUE is a standard collection of natural-language understanding benchmarks (Wang et al., 2019); we use SST-2 sentiment classification, MRPC paraphrase classification, and RTE textual entailment. The setting is intentionally compressed—512 training examples and two epochs—so it should be read as a stress test of attribution, not as a definitive ranking of PEFT methods.

The central contribution is an evaluation insight: under modest vanilla controls, the apparent source of improvement changes. Rank and learning-rate sweeps substantially alter the conclusions drawn from default-only comparisons, while raw prediction checks expose instability that averages would hide. We therefore propose a lightweight reporting protocol that makes baseline tuning and failures visible without requiring an unlimited hyperparameter search.

2 Related Work

PEFT and LoRA. Adapters introduced a compact alternative to full fine-tuning (Houlsby et al., 2019). LoRA later represented updates with low-rank factors while freezing pretrained weights (Hu et al., 2022).

Rank allocation. AdaLoRA allocates a rank budget using parameter importance and prunes singular directions during training (Zhang et al., 2023). Related work studies flexible, salience-based, and gradient-driven rank allocation (Ke et al., 2024; He et al., 2025; Liu et al., 2026). If rank choice is central enough to motivate a new mechanism, a small fixed-rank sweep is a natural attribution control.

LoRA optimization. LoRA-RITE proposes a transformation-invariant optimizer for low-rank factors (Yen et al., 2025). LoRA-Pro and ALLoRA likewise show that optimization and learning-rate behavior are active design questions (Wang et al., 2025; Huang and Balestrieri, 2024). Our question is complementary: does a specialized optimizer remain advantageous when vanilla LoRA with AdamW receives a small rank/LR sweep?

3 Experimental Setup

We use RoBERTa-base (Liu et al., 2019) with LoRA-style PEFT. Metrics follow common GLUE practice: accuracy for SST-2 and RTE, and F1 for MRPC. Every run uses 512 training examples, 500 validation examples, 2 epochs, batch size 16, and the same base-model cache.

Dynamic-rank audit. We compare fixed LoRA ranks $r \in \{4, 8, 16, 32\}$ with AdaLoRA target ranks $\{8, 16\}$. The default fixed baseline is $r = 8$. The seven dataset-seed cells are SST-2 seeds 42/43, MRPC seeds 42/43, and RTE seeds 42/43/44.

Optimizer audit. We compare AdamW LoRA and LoRA-RITE at ranks 8 and 16. AdamW uses learning rates $\{10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}\}$; LoRA-RITE uses $\{10^{-4}, 2 \times 10^{-4}\}$. The default AdamW baseline is $r = 8$, learning rate 2×10^{-4} . The six cells are SST-2, MRPC, and RTE crossed with seeds 42/43.

Why the search spaces are asymmetric. Fixed LoRA receives four ranks while AdaLoRA receives two target ranks; AdamW receives six rank/LR configurations while LoRA-RITE receives four. We did not add post-hoc trials to equalize grid density because the original study was designed to ask a narrower audit question: can a modest vanilla control change attribution? The tuned baseline is therefore a diagnostic control, not a compute-matched tournament. Equal-density or equal-compute tuning remains necessary before ranking the methods themselves.

4 Results

4.1 Default baselines look more favorable

AdaLoRA beats fixed LoRA at the default rank in 4/7 cells, but the best tested fixed rank in only 1/7. LoRA-RITE beats default AdamW in 3/6 cells, but tuned AdamW in only 1/6. These counts describe the tested cells; they are not precise population win

rates. Wilson 95% intervals are correspondingly wide: 1/7 gives 0.03–0.51 and 1/6 gives 0.03–0.56.

The mean AdamW rank/LR sweep gain over the default is +0.1055, whereas the mean LoRA-RITE effect is -0.0013 against default AdamW and -0.1068 against tuned AdamW. For AdaLoRA, the mean effect is -0.0963 against fixed $r = 8$ and -0.1102 against the best tested fixed rank.

4.2 Rank sweeps shrink adaptive-rank gains

AdaLoRA sometimes improves over $r = 8$, especially outside SST-2. However, the best fixed rank usually matches or exceeds the best AdaLoRA target. The surviving nonnegative margins against the tuned rank are small outside the collapsed SST-2 cells (Table 2). This does not show that adaptive rank is useless; it shows that a single fixed rank is insufficient evidence for attributing gains to adaptive allocation.

4.3 Learning-rate sensitivity and compressed data

The optimizer result is not a cosmetic consequence of selecting among nearly identical settings. On SST-2, AdamW at 5×10^{-4} reaches 0.8920/0.8780 accuracy, whereas the default AdamW cells are 0.6980/0.5360. The compressed regime can amplify optimization sensitivity: with only 512 training examples and two epochs, some configurations fail to leave the majority-class solution. We therefore interpret the result as evidence that default-only attribution is fragile under this budget, not as an explanation of sensitivity in larger-scale LLM tuning.

4.4 Sanity check: label collapse, not a parser bug

We reran eight suspicious SST-2 cells with raw prediction export. Saved metrics match raw prediction accuracy in every sanity run. Four AdaLoRA runs and one LoRA-RITE run predict label 1 for all 500 validation examples, matching the 265/500 majority-class baseline and producing 0.5300 accuracy. The tuned AdamW controls do not collapse and reach 0.8920/0.8780. These are observed failures under the tested budget, not universal properties of the methods.

5 Recommendations

A minimal PEFT comparison should: (1) declare a small vanilla rank/LR sweep before examining outcomes; (2) report default and tuned controls; (3)

Audit	Method	Search grid	Learning rate / optimizer	Runs
Rank	Fixed LoRA	$r = \{4, 8, 16, 32\}$	8×10^{-4} / AdamW	28
Rank	AdaLoRA	target $\{8, 16\}$	8×10^{-4} / AdamW	14
Optimizer	AdamW LoRA	$r = \{8, 16\}$, LR $\{1, 2, 5\} \times 10^{-4}$	AdamW	36
Optimizer	LoRA-RITE	$r = \{8, 16\}$, LR $\{1, 2\} \times 10^{-4}$	LoRA-RITE	24

Table 1: Search budget. The baseline sweep is an attribution control; the grids are intentionally small and not symmetric.

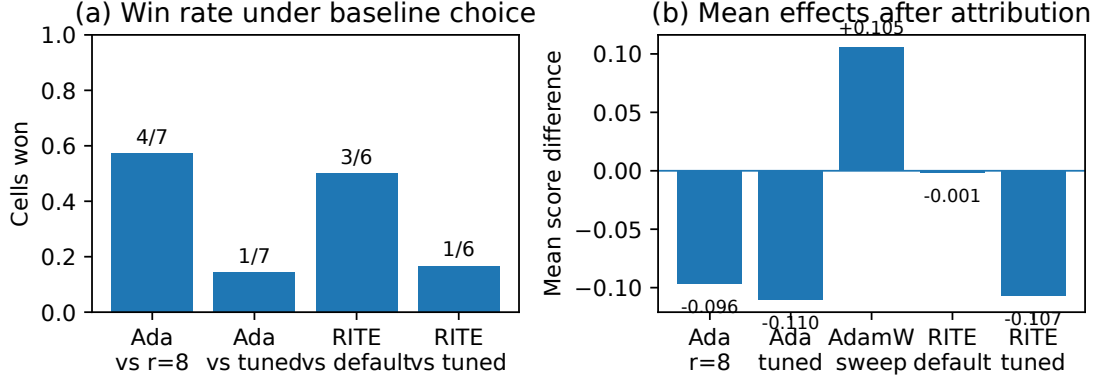


Figure 1: Headline audit. Specialized variants look more favorable against default vanilla settings than against modestly tuned controls. The AdamW sweep gain is itself larger than the mean LoRA-RITE effect against default AdamW.

Cell	AdaLoRA–best fixed	RITE–best AdamW
MRPC-42	+0.0050	+0.0034
MRPC-43	-0.0001	-0.0005
RTE-42	-0.0036	-0.0144
RTE-43	-0.0144	-0.0614
RTE-44	+0.0000	–
SST-2-42	-0.3820	-0.3620
SST-2-43	-0.3760	-0.2060

Table 2: Per-cell margins against tuned hidden baselines. Positive values mean the specialized method wins.

show per-cell margins and seed variation, rather than only averages; (4) disclose whether search is equalized by run count or compute; and (5) preserve unstable cells with prediction distributions instead of silently filtering them. The protocol is not a mandate for unlimited tuning. Its purpose is to prevent a cheap missing control from being mistaken for a method-specific gain.

6 Conclusion

A small, method-agnostic sweep over vanilla LoRA can materially change the attribution of apparent PEFT gains in this low-budget setting. The result is not a ranking of AdaLoRA or LoRA-RITE; it is evidence that default-only baselines are too fragile to support strong mechanism claims without modest tuning and failure diagnostics.

Limitations

This study uses capped GLUE subsets, RoBERTa-base, encoder classification, limited seeds, and only three task families. GLUE is mature rather than current, but its small, standardized tasks make failure auditing and raw-prediction verification tractable; results should not be extrapolated to modern instruction tuning without replication. The study is not a full reproduction of AdaLoRA or LoRA-RITE at their original scale. Search spaces are small and asymmetric, and the number of cells is too small for strong statistical claims. Some gaps may reflect robustness under compressed training rather than intrinsic method quality. Peak GPU memory was not logged for every run. Most importantly, the result supports a reporting recommendation, not a broad dismissal of specialized PEFT methods.

Acknowledgments

We thank the Insights 2026 reviewers for suggestions that improved the framing of the small- n evidence, clarified the asymmetric search budget, and motivated a fuller discussion of learning-rate sensitivity and the compressed training regime.

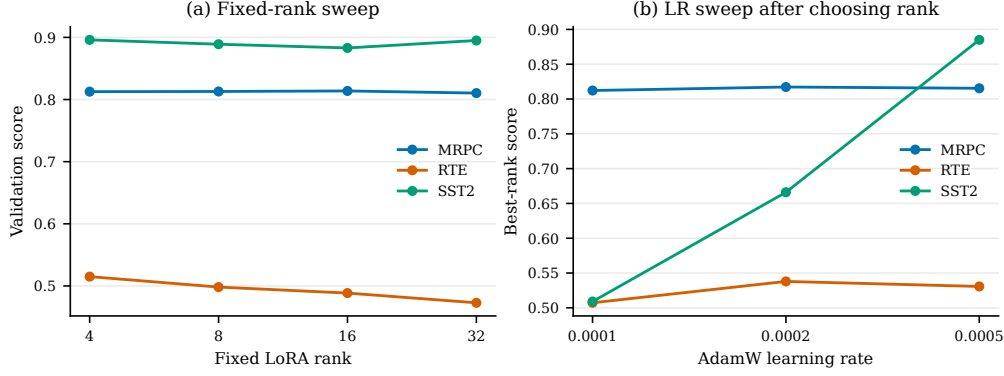


Figure 2: Hidden baselines come from small, method-agnostic sweeps. Left: changing fixed LoRA rank changes the baseline. Right: AdamW learning-rate tuning, after selecting the best tested rank at each learning rate, changes the optimizer comparison.

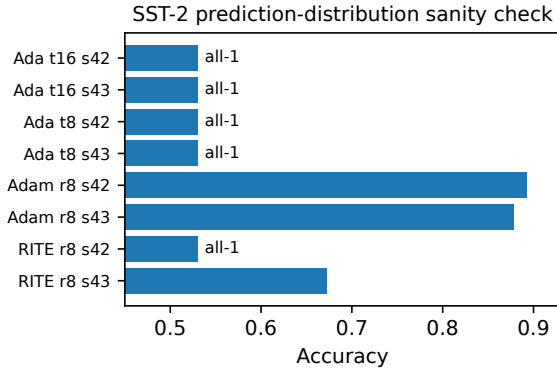


Figure 3: SST-2 sanity audit. Runs marked “all-1” predict the positive label for all 500 validation examples.

References

- Haonan He, Peng Ye, Yuchen Ren, Yuan Yuan, Luyang Zhou, Shucun Ju, and Lei Chen. 2025. GoRA: Gradient-driven adaptive low rank adaptation. arXiv:2502.12171.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of ICML*, pages 2790–2799.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*.
- Hai Huang and Randall Balestriero. 2024. ALLoRA: Adaptive learning rate mitigates LoRA fatal flaws. arXiv:2410.09692.
- Wenjun Ke, Jiahao Wang, Peng Wang, Jiajun Liu, Dong Nie, Guozheng Li, and Yining Li. 2024. Unveiling LoRA intrinsic ranks via salience analysis. In *NeurIPS*.
- Muqing Liu, Chongjie Si, and Yuheng Jia. 2026. FlexLoRA: Entropy-guided flexible low-rank adaptation. In *ICLR*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.
- Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. 2025. LoRA-Pro: Are low-rank adapters properly optimized? In *ICLR*.
- Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Sai Surya Duvvuri, Inderjit S. Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. 2025. LoRA done RITE: Robust invariant transformation equilibration for LoRA optimization. In *ICLR*.
- Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*.

A Full Dynamic-Rank Cell Table

Task	Seed	$r = 8$	Best fixed	Best r	Best AdaLoRA	Ada- $r = 8$	Ada-best
MRPC	42	.8173	.8173	8	.8223	+0.0050	+0.0050
MRPC	43	.8086	.8154	16	.8152	+0.0067	-0.0001
RTE	42	.5199	.5307	4	.5271	+0.0072	-.0036
RTE	43	.5018	.5415	4	.5271	+0.0253	-.0144
RTE	44	.4729	.4729	4	.4729	+0.0000	+0.0000
SST-2	42	.8920	.9120	4	.5300	-.3620	-.3820
SST-2	43	.8860	.9060	32	.5300	-.3560	-.3760

Table 3: Full dynamic-rank comparison. MRPC uses F1; SST-2 and RTE use accuracy.

B Full LoRA-RITE Cell Table

Task	Seed	Default AdamW	Tuned AdamW	Tuned config	Best RITE	RITE-default	RITE-tuned
MRPC	42	.8122	.8223	$r = 16, LR=2e-4$	.8257	+0.0135	+0.0034
MRPC	43	.8122	.8127	$r = 8, LR=5e-4$	.8122	+0.0000	-.0005
RTE	42	.5415	.5415	$r = 8, LR=2e-4$	.5271	-.0144	-.0144
RTE	43	.5018	.5884	$r = 8, LR=5e-4$	.5271	+0.0253	-.0614
SST-2	42	.6980	.8920	$r = 8, LR=5e-4$	.5300	-.1680	-.3620
SST-2	43	.5360	.8780	$r = 8, LR=5e-4$	.6720	+0.1360	-.2060

Table 4: Full LoRA-RITE comparison. MRPC uses F1; SST-2 and RTE use accuracy.

C SST-2 Sanity Details

Run	Acc.	Pred 0	Pred 1	Gold 0	Gold 1	Assessment
AdaLoRA target 8 seed 42	.5300	0	500	235	265	label collapse
AdaLoRA target 8 seed 43	.5300	0	500	235	265	label collapse
AdaLoRA target 16 seed 42	.5300	0	500	235	265	label collapse
AdaLoRA target 16 seed 43	.5300	0	500	235	265	label collapse
AdamW tuned seed 42	.8920	213	287	235	265	metric pipeline sane
AdamW tuned seed 43	.8780	244	256	235	265	metric pipeline sane
LoRA-RITE seed 42	.5300	0	500	235	265	label collapse
LoRA-RITE seed 43	.6720	85	415	235	265	not full collapse

Table 5: SST-2 prediction-distribution sanity check. The saved metric matches raw prediction accuracy in every retained row.

D Budget Accounting

The budget accounting behind Table 1 records audit family, task, seed, method, rank or target rank, optimizer, learning rate, trainable parameters, training time, and peak-memory status. Peak GPU memory is reported only when present in the recovered source results; missing memory values are not imputed. This preserves the distinction between measured resource use and unavailable logging.

E Additional Figures

The following diagnostics reproduce the auxiliary views used in the accepted version. They are descriptive checks rather than additional experiments.

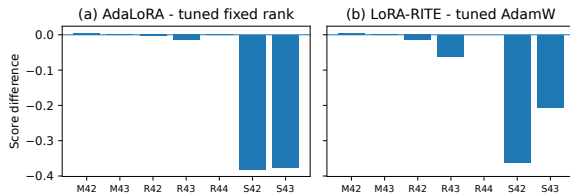


Figure 4: Cell-level score differences against tuned hidden baselines. Positive values mean the specialized method wins the dataset-seed cell after the vanilla baseline sweep.

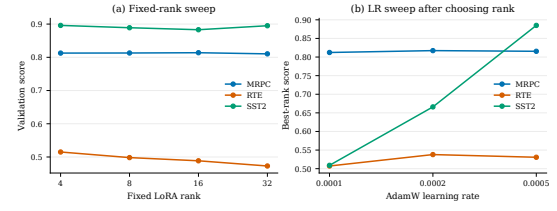


Figure 5: Rank and learning-rate sweeps used to define the tuned hidden baselines.

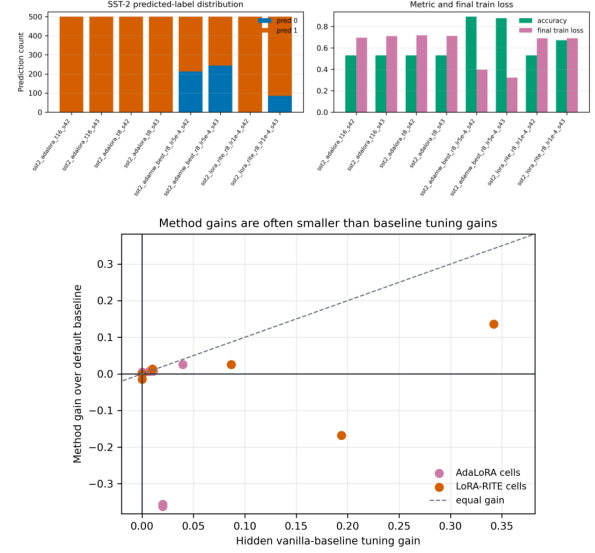


Figure 6: SST-2 failure diagnostics and the relation between method gain and hidden-baseline tuning gain.

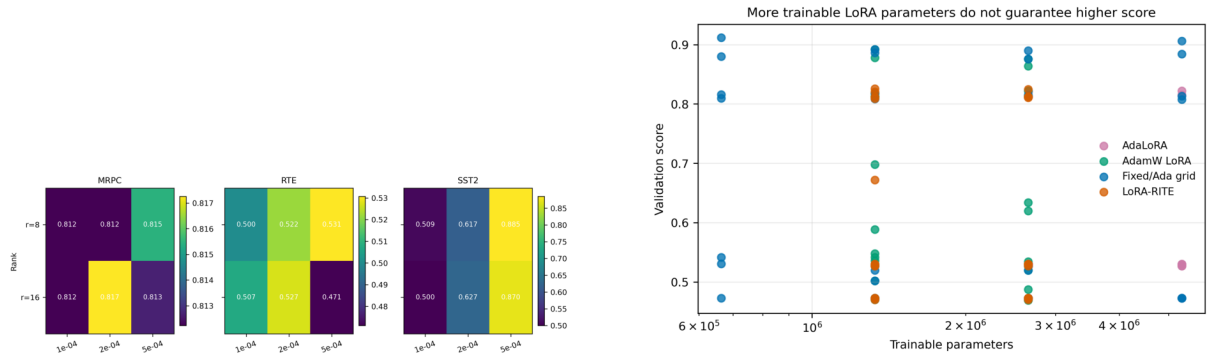


Figure 7: Additional diagnostic views for the optimizer grid and parameter budget. Learning-rate sensitivity is visible, while trainable-parameter count alone does not determine validation score.

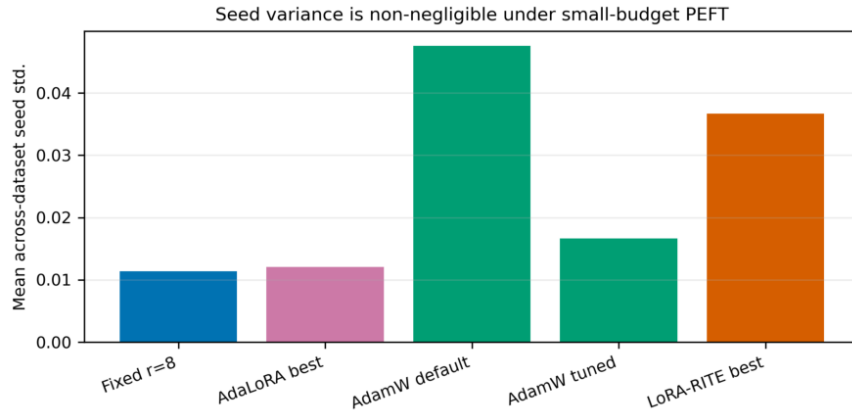


Figure 8: Seed variance is non-negligible under the small-budget PEFT setting, motivating cell-level and seed-aware reporting alongside means.

When Relation-Free Retrieval Is Not Relation-Aware: A Directionality Stress Test for Linear Graph RAG

Ding Wu

Independent Researcher

dingwu@alumni.upenn.edu

Abstract

Relation-free graph retrieval avoids explicit relation extraction, but removing relation labels can weaken shallow ranking when argument order determines which evidence is correct. We test this boundary condition with the official LinearRAG retrieval-only pipeline. On a controlled set of 3,000 questions over nine asymmetric relation types, LinearRAG recovers the gold passage at rank 20 in 96.4% of cases but ranks it first in only 18.2%. Its top-1 output is a same-entity wrong-relation passage in 1,481 cases and a reversed-relation passage in 780 cases, for 2,261 direction errors (75.4%); a further 108 cases are same-relation wrong-entity errors and are reported separately. A local Qwen2.5 reader shows that shallow retrieval errors mainly propagate as abstention or answer failure. The result is deliberately bounded: it diagnoses first-stage retrieval under asymmetric ambiguity, not the strongest possible reranked GraphRAG system. We recommend relation-sensitive shallow-ranking and reader-propagation checks in addition to aggregate recall.

1 Introduction

Graph-based retrieval-augmented generation (GraphRAG) organizes documents around entities, relations, and paths so that retrieval can move beyond flat lexical or dense nearest-neighbor matching (Edge et al., 2024; Chen et al., 2025; Hu et al., 2024). Explicit relation extraction, however, can be expensive, noisy, and difficult to maintain. LinearRAG removes typed relations and constructs a relation-free graph over text and entities (Zhuang et al., 2025).

This simplification has a natural boundary condition: asymmetric relations, where subject–object order determines the correct evidence. “Alice acquired BetaCorp” and “BetaCorp acquired Alice’s supplier” share entities and relation words, but only the first supports “Who acquired BetaCorp?” A

relation-free retriever can reach the correct neighborhood while still placing a directionally wrong passage first.

We study that failure as an Insights-style diagnostic rather than proposing a new method. The questions are: (1) is gold evidence absent or merely under-ranked; (2) are shallow failures random or relation-specific; (3) does the pattern persist under paraphrase and on a small real-data slice; and (4) do shallow retrieval errors propagate to a reader? We evaluate the official LinearRAG index and retrieval interface, not full answer generation or a production reranking stack.

2 Background

Relation-free GraphRAG. LinearRAG builds a relation-free Tri-Graph and activates entities from a question, aggregating global importance to rank passages without extracted relation triples (Zhuang et al., 2025). This removes a major construction cost, but the retrieval representation does not explicitly encode argument direction.

Asymmetric relations. Relations such as acquired, defeated, replaced, parent of, and part of are not invariant to argument order. A passage can contain the correct entities and relation cue while supporting the opposite answer. Aggregate Recall@20 can therefore hide a shallow-ranking failure that matters when a reader consumes only the first few passages.

GraphRAG design space. Other graph-RAG systems retain more explicit structure, for example relational paths, query-centric graph chunks, or domain-specific graph expansion (Chen et al., 2025; Wu et al., 2025; Mostafa et al., 2024). These systems are not our comparison target; they illustrate the broader construction–ranking trade-off that makes LinearRAG a useful diagnostic case.

Evaluation context. Standard RAG evaluation emphasizes answer quality or relevant-passage recall (Lewis et al., 2020). GraphRAG benchmarks broaden system-level comparisons (Xiang et al., 2025); our study isolates one mechanism-level diagnostic. Qwen2.5 is used only as a local reader, not as an external judge (Qwen Team, 2025).

3 Experimental Setup

3.1 Controlled stress set

The controlled set contains 3,000 questions and 15,000 passages. Each question is paired with five candidates: one *gold passage*, defined as the unique passage that states the queried directed relation with the correct arguments; one reversed-relation hard negative; one same-entity wrong-relation hard negative; one same-relation different-entity hard negative; and one irrelevant passage. Thus all hard negatives are plausible candidates, but only the gold passage entails the directed answer.

The set spans acquired, founded, defeated, replaced, caused, parent, CEO, located, and part-of. Half the questions use relation-specific templates and half are cached paraphrases. Archived experiment records preserve the template/paraphrase split and aggregate outcomes, but not the exact paraphrase-generation prompt. We therefore treat the paraphrase comparison as a secondary robustness check rather than a fully reproducible generation procedure.

3.2 Real-data slice

We also evaluate 300 public examples from **2Wiki-MultiHopQA** (2Wiki) over 3,323 chunks. A gold passage is a chunk whose title belongs to the benchmark supporting-fact titles for the query. The slice is strongly imbalanced: CEO 118, parent 94, located 84, and founded 4. It is used only as a qualitative transfer check; we make no relation-level claim for founded.

3.3 Retrievers and reader

We evaluate a local clone of the official LinearRAG pipeline at commit 5da5262, calling `index()` and `retrieve()` but not `qa()`. Diagnostic baselines separate entity overlap, lexical cues, dense similarity, and direction-sensitive argument order. Entity-only ranks by overlapping extracted entities. BM25 uses tokenized passages. Dense retrieval uses cosine similarity from all-MiniLM-L6-v2. Hybrid combines per-query min-max normalized

BM25 and dense scores with $\alpha \in \{0.25, 0.5, 0.75\}$. Relation-aware lexical adds fixed relation-cue, expected-argument, and reversed-template indicators to BM25. These baselines are diagnostic controls, not proposed replacements.

Reader propagation uses a local Qwen2.5-7B-Instruct model. It receives only the supplied context and is instructed to answer “unknown” when unsupported. The 300-example stress reader comparison covers LinearRAG top-1, top-5, top-20, Hybrid top-5, and a gold-passage oracle. A 50-example real-slice reader check is retained only as an artifact-level sanity check because long real contexts caused frequent truncation. No external API is used. Retrieval metrics are R@1, R@3, R@5, R@20, MRR, mean gold rank, wrong-relation error rate, reversed-relation error rate, and entity-trap rate; reader metrics include exact match, contains match, abstention, wrong/reversed answer rates, latency, and context length.

4 Results

4.1 Gold evidence is usually under-ranked

Table 1 shows the main boundary condition. LinearRAG reaches R@20 of 0.964, but R@1 is 0.182 and R@5 is 0.434. The gold passage is therefore commonly available to a deep retriever but absent from the shallow context a reader is likely to receive. Hybrid $\alpha = .25$ reaches R@1 of 0.575 and R@20 of 0.988, indicating that lexical and dense query evidence can repair part of the shallow ordering.

4.2 Top-1 failures are relation traps

The top-1 taxonomy contains 1,481 same-entity wrong-relation outputs and 780 reversed-relation outputs. These sum to **2,261 direction errors** (75.4%). The remaining 108 same-relation wrong-entity outputs measure entity discrimination and are reported separately. Only 85 top-1 outputs are irrelevant, while 546 are gold. The dominant failure is therefore relation ambiguity rather than random drift.

4.3 Paraphrase and real-data checks

Templated and paraphrased questions show similar shallow behavior in the archived aggregate results, although the missing exact paraphrase prompt limits reproducibility of this comparison. Relation-level variation is more informative: defeated and

Method	R@1	R@3	R@5	R@20	MRR	Rev. err	Wrong-rel. err	Entity trap
Official LinearRAG	.182	.297	.434	.964	.317	.260	.494	.756
Entity-only proxy	.142	.330	.469	.794	.292	.098	.659	.766
BM25	.436	.638	.706	.910	.557	.270	.223	.535
Dense MiniLM	.467	.664	.748	.950	.593	.319	.080	.402
Hybrid $\alpha = .25$	.575	.764	.849	.988	.693	.329	.061	.395
Hybrid $\alpha = .50$	.571	.746	.841	.990	.683	.241	.160	.410
Hybrid $\alpha = .75$	.515	.698	.802	.979	.633	.237	.210	.464
Relation-aware lexical	.465	.694	.791	.970	.601	.254	.219	.515

Table 1: Controlled direction-stress retrieval results on 3,000 questions. LinearRAG often reaches gold by rank 20 but under-ranks it shallowly.

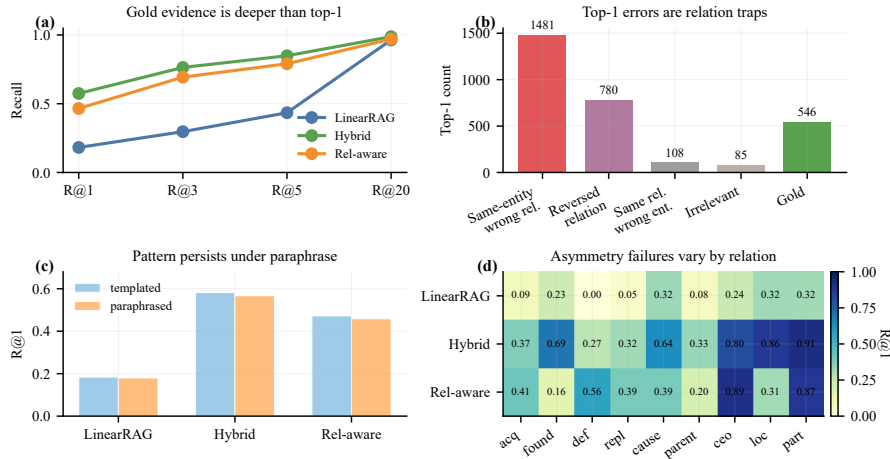


Figure 1: Directionality diagnostics. (a) LinearRAG reaches gold by rank 20 but under-ranks it shallowly. (b) Top-1 outcomes concentrate in same-entity wrong-relation and reversed-relation passages; the 108 same-relation wrong-entity cases are not direction errors. (c) Templated and paraphrased questions show similar shallow behavior in the archived aggregate results. (d) Relation types vary sharply, with defeated and acquired especially brittle.

Method	R@1	R@3	R@5	R@20	MRR
LinearRAG	.470	.587	.680	.750	.553
Entity-only	.197	.300	.333	.403	.258
BM25	.727	.847	.887	.947	.800
Dense	.347	.423	.467	.557	.400
Hybrid $\alpha = .75$	.743	.857	.897	.950	.811
Rel.-aware	.723	.857	.880	.947	.799

Table 2: Real 2Wiki relation slice (300 examples). This imbalanced slice is a qualitative check only.

acquired are brittle, whereas CEO and located retain stronger lexical anchors.

On the 300-example 2Wiki slice, LinearRAG reaches R@1 of 0.470 and R@20 of 0.750. BM25 and relation-aware lexical reach R@1 of 0.727 and 0.723. This result should not be interpreted as a broad superiority claim for BM25. The slice is small, relation-imbalanced, and lexically favorable: query and support text often preserve explicit entity and relation strings. It is a boundary check showing that literal overlap can be unusually strong in this construction.

4.4 Shallow errors propagate to the reader

With LinearRAG top-1 context, the reader obtains a contains-match rate of 0.227 and abstains in 0.760 of cases. Top-5 improves only modestly to 0.290. LinearRAG top-20 reaches 0.550 but uses roughly 425 context tokens, compared with 101 for top-5. Hybrid top-5 reaches 0.497 at a similar context length, while the gold oracle reaches 0.953. Wrong-relation answer rates remain low; failures are mainly abstention or an unsupported answer.

This single-reader experiment is a propagation diagnostic, not evidence that all readers behave identically. A second reader was not available in the surviving experiment package, so model-specific behavior remains a limitation.

5 Discussion

What the result does and does not show. The study does not show that relation-free GraphRAG is generally ineffective. LinearRAG removes a costly relation-extraction step, and high R@20 shows that it often reaches the correct evidence neighborhood.

The narrower result is that entity-sharing but directionally wrong passages can be ranked above gold under asymmetric ambiguity.

Where reranking fits. A lightweight cross-encoder or LLM reranker could score query–passage pairs after LinearRAG candidate generation and explicitly compare argument roles. Such a stage may preserve the construction efficiency of a relation-free graph while restoring shallow precision. However, reranking cannot recover a gold passage that is absent from the candidate pool, and it adds latency and model dependence. The present experiment isolates the first-stage ranking; it should not be read as the performance ceiling of a full production pipeline.

Diagnostic interpretation. Entity-only, lexical, dense, hybrid, and relation-aware lexical baselines separate three signals: entity overlap, surface relation cues, and direction-sensitive argument order. Their role is mechanistic diagnosis, not a new system claim.

6 Conclusion

Relation-free graph retrieval can be efficient and practical, but it is not automatically relation-aware when argument order matters. In the official LinearRAG retrieval-only setup, gold evidence is often recovered at deeper ranks while same-entity wrong-relation and reversed-relation passages dominate shallow errors. A local reader analysis shows that these errors propagate mainly as answer failure or abstention. Relation-free GraphRAG evaluations should therefore report relation-sensitive shallow-ranking diagnostics and reader propagation, not only aggregate recall.

Limitations

The controlled set deliberately amplifies asymmetric direction errors and does not estimate their prevalence in natural corpora. The real-data check is 2Wiki-only and relation-imbalanced. Archived experiment artifacts preserve the split and aggregate results but not the exact paraphrase-generation prompt or original generation script. We therefore do not claim a complete reproducibility release for the stress set, and the paraphrase result should be treated as secondary. We did not evaluate a cross-encoder or LLM reranker, so the claim concerns the tested first-stage retrieval rather than the strongest possible production pipeline. The reader analysis

uses one local Qwen2.5 model, and multi-reader validation remains future work.

Acknowledgments

We thank the Insights 2026 reviewers for identifying the error-category arithmetic issue and for prompting clearer definitions of gold evidence, 2WikiMultiHopQA, reranking scope, synthetic-data construction, and single-reader limitations.

References

- Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. 2025. PathRAG: Pruning graph-based retrieval augmented generation with relational paths. arXiv:2502.14902.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Metropolitansky. 2024. From local to global: A Graph RAG approach to query-focused summarization. arXiv:2404.16130.
- Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. GRAG: Graph retrieval-augmented generation. arXiv:2405.16506.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*.
- Radeen Mostafa, Mirza Nihal Baig, Mashaekh Tausif Ehsan, and Jakir Hasan. 2024. G-RAG: Knowledge expansion in material science. arXiv:2411.14592.
- Qwen Team. 2025. Qwen2.5 technical report. Technical report.
- Yaxiong Wu, Jianyuan Bo, Yongyue Zhang, Sheng Liang, and Yong Liu. 2025. Query-centric graph retrieval augmented generation. arXiv:2509.21237.
- Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jin-song Su. 2025. When to use graphs in RAG: A comprehensive analysis for graph retrieval-augmented generation. arXiv:2506.05690.
- Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. 2025. LinearRAG: Linear graph retrieval augmented generation on large-scale corpora. arXiv:2510.10114.

Author Index

- Abraham, Ashish, 1
Arun, Govind, 1
- Barak, Libby, 19
Bouneffouf, Djallel, 43
- Chen, Han, 90
- Datar, Mayur, 48
De Marneffe, Marie-Catherine, 33
Delgado, José Luis, 115
Dershowitz, Gabriel, 43
Dey, Soumyadeep, 48
- Feldman, Anna, 19
- Hu, Julia, 70
Huber, Marius, 7
- Kazakov, Allan, 102
Kumar, Gaurav, 60
- Lakshmipathi, Kumar, 70
Lee, Patrick, 19
Li, Yingrui, 90
Loftus, Sebastian, 33
- Merrett, Geoff, 76
- Middleton, Stuart, 76
Mishra, Bamdev, 48
- Niranjan, Shobhit, 48
- Opitz, Juri, 7
Osman, Yaseen, 76
- P, Gadha Lekshmi, 1
Peng, Jing, 19
Plank, Barbara, 33
Pletenev, Sergey, 120
Poh, Whitney, 19
- Sammartino, Julia, 19
Sawant, Yash, 110
Shen, Alfred, 70
Singh, Prabhjot, 84
Soykan, Gürkan, 95
Şahin, Gözde, 95
- Takahashi, Noriko, 19
Tenneti, Srikanth, 48
- Wang, Xinpeng, 33
Wu, Ding, 127, 133