Prompt Rewrites Are Semantic Actions: Testing Whether Language Models Obey Typed Operator Laws

Zida Yang[1]* **Weiqing Zeng**[2] **Guquan You**[3,4] **Yu Zheng**[2,5]†
[1]Guanghua School of Management, Peking University, Beijing, China
[2]School of Computing and Artificial Intelligence,
Southwest Jiaotong University, Chengdu, China
[3]Institute of Emergency Safety Innovation, Guangzhou University, Guangzhou, China
[4]Guangdong Guangda Mingxin Technology Co., Ltd. [5]JD.COM
2401111090@stu.pku.edu.cn, zengweiqing@my.swjtu.edu.cn
youguquan@gzhu.edu.cn, msyuzheng@outlook.com

Abstract

Prompt rewrites can prescribe a change in an answer, not merely a change in wording. We introduce ORBIT-SIT, a framework that assigns each rewrite an explicit action on a task's output space and separately measures correctness against transformed gold answers and consistency with transformed model predictions. Across eight operators, five datasets, and three instruction-tuned models, high task accuracy does not ensure compliance: Phi-4-mini achieves 90.7% accuracy on SST-2 but only 16.4% under a label-swap instruction. Deterministic validation and a three-annotator audit check the operator contracts. Non-deceptive binary controls retain a 17.8-percentage-point gap, and separating inference from transformation improves compliance, implicating joint execution rather than refusal alone. OrbitAlign, a quantized low-rank adaptation objective, reduces gold-grounded risk by 25.4 points on a 400-item BoolQ repair pool. The recipe lowers risk in 39 of 40 cells on two additional backbones, and same-action transfer across binary tasks reduces risk by 12.9–15.1 points. The binary tasks provide the primary evidence; structured tasks remain extensibility diagnostics. ORBIT-SIT makes declared prompt actions measurable and partly repairable without equating compliance with general understanding.

1 Introduction

Consider a sentiment classifier that correctly labels "A delightful film" as positive. Now keep the review fixed and change the instruction: "Return the opposite label: negative for positive sentiment and positive for negative sentiment." The required output is negative. Returning positive is stable across prompts, yet violates the requested mapping. Conversely, a model that initially misclassifies the review as negative and then returns positive follows the mapping consistently while remaining wrong about the transformed gold answer. These cases require two questions: *Is the rewritten answer correct?* and *Does it apply the requested action to the model's own clean answer?*

This distinction matters when a language model is reused through different prompts, output schemas, and interfaces. Multi-prompt evaluation and robustness testing show that instruction wording can change model behavior (Mizrahi et al., 2024; Zhu et al., 2024). However, stability is the appropriate target only when the rewrite preserves the answer. A test of an answer-changing instruction must specify how the answer should change.

We introduce **ORBIT-SIT**, an evaluation framework in which a prompt rewrite carries an explicit, typed action on the task output. The action may preserve a label, swap a binary label, or map a rich output to a simpler space, such as event types without trigger spans. Two risks compare the rewritten prediction with the transformed gold answer and with the transformed clean prediction, respectively. Figure 1 illustrates this procedure. Their joint use separates transformed-task correctness from consistency; neither quantity alone establishes general semantic understanding.

We instantiate eight operators on five datasets with three open-weight instruction-tuned models. The clearest evidence comes from SST-2 and BoolQ, where outputs and actions are binary. For example, Phi-4-mini's 90.7% SST-2 accuracy contrasts with 16.4% correctness under the label-swap operator, a 74.3-percentage-point gap. Structured tasks extend the scoring spaces but often lack enough correct clean answers for precise conditional estimates; we report them as stress tests.

*Project leader.
†Corresponding author.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 2649–2671
November 6-10, 2026 ©2026 Association for Computational Linguistics

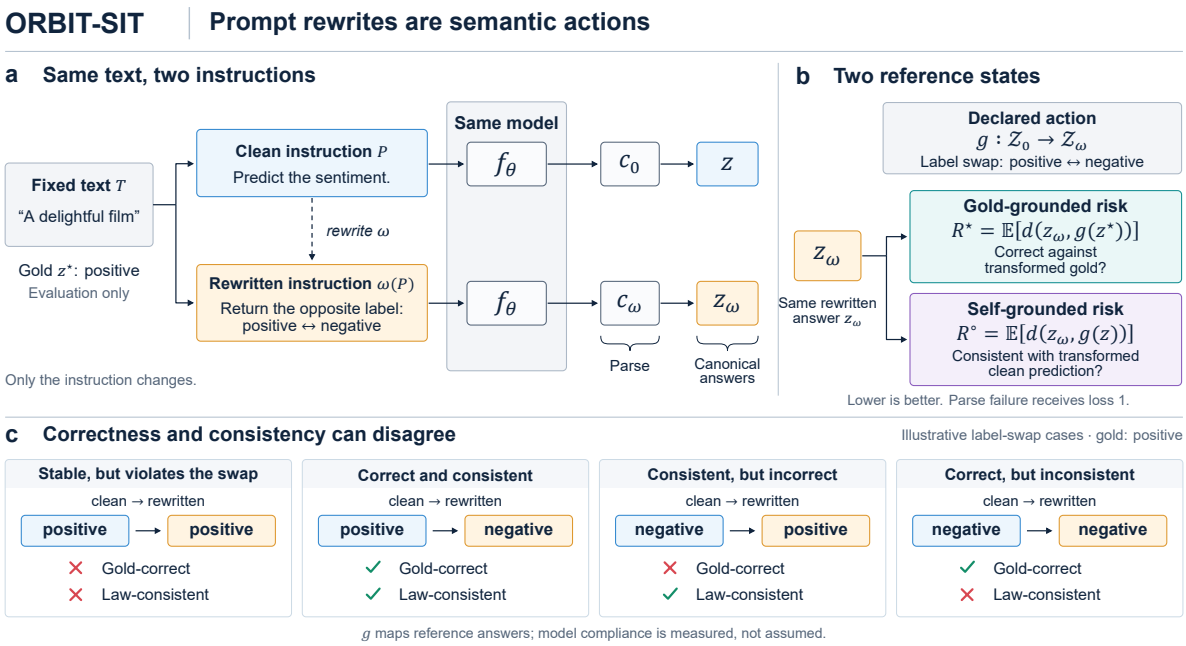


Figure 1: ORBIT-SIT in a binary label-swap example. (a) The text stays fixed while clean and rewritten instructions pass through the same model and their respective canonicalizers. (b) The rewritten answer is compared with transformed gold and with the transformed clean prediction; gold is used only for evaluation. (c) Four illustrative outcomes compare rewritten-answer correctness against transformed gold with law consistency. The task index τ is omitted in the diagram: c_0, c_ω, and g denote the task-specific maps. These examples are not additional experimental items.

An instruction to output an opposite label can conflict with a model's ordinary answering policy. We therefore distinguish explicit refusal from other non-compliance and evaluate neutral codebooks, separate ordinary-answer and operator-output fields, and an operator-only control. The residual gap survives these non-deceptive conditions. One-call decomposition helps, and a two-call infer-then-transform pipeline performs better still. These controls locate the difficulty in joint execution under the tested interface, without treating a refusal as evidence of cognitive inability.

Contributions.

- A typed operator protocol with separate gold-grounded and self-grounded risks, deterministic contract validation, and a human semantic audit. The full operator criterion is strictly stronger than invariance-only stability.
- Controlled evidence that high binary-task accuracy can coexist with operator-contract non-compliance, including non-deceptive and decomposed-inference baselines that qualify its interpretation.
- **OrbitAlign**, a proof-of-concept repair using quantized low-rank adaptation (QLoRA; Dettmers et al., 2023). We evaluate disjoint items, unseen wordings of the same action, replication across backbones, and same-action transfer across tasks. Representation and patching analyses provide additional diagnostics rather than a claim of circuit localization.

2 Related Work

Prompt robustness and behavioral testing. Multi-prompt evaluation (Mizrahi et al., 2024), punctuation perturbations (Seleznyov et al., 2025), component perturbations (Zheng et al., 2026), PromptBench (Zhu et al., 2024), dynamic text-embedding evaluation (Frank and Afli, 2026), and prediction benchmarks (Habba et al., 2025) study variation in model performance. CheckList organizes behavioral capabilities and tests (Ribeiro et al., 2020); metamorphic testing evaluates relations between transformed inputs and outputs (Cho et al., 2025). ORBIT-SIT builds on this relational view. It fixes the task text, declares an action on the output space, and reports correctness and self-consistency separately. Table 1 summarizes the distinction at the level of evaluation targets, rather than claiming that earlier work lacks all transformation tests.

Evaluation view	Principal comparison
Prompt averaging	Task score across prompt variants
Invariance testing	Output before/after a preserving rewrite
Metamorphic testing	A specified input–output relation
ORBIT-SIT	Transformed gold and transformed clean prediction, separately

Table 1: Related evaluation targets. ORBIT-SIT instantiates a metamorphic relation with typed prompt actions and dual reference states. This is a conceptual comparison, not a benchmark ranking.

Neighboring notions of semantic reliability. Semantic-invariance testing of agentic models (de Zarzà et al., 2026) studies meaning-preserving transformations. Logical robustness on structured satisfiability problems (Es-sebbani et al., 2026) and dictionary-sense semantic evaluation (Zubillaga et al., 2026) assess different aspects of semantic reliability. These motivate careful operational definitions, but they are not evidence that a model refuses our particular label-swap instructions. Our contribution is the typed, auditable evaluation protocol, not a claim to have first discovered instruction sensitivity. Likewise, self-consistency decoding (Wang et al., 2023) aggregates reasoning paths; our self-grounded risk instead compares predictions across a declared action.

Diagnostics, equivariance, and adaptation. Prompt formatting can affect hidden representations (Kang et al., 2025); linear probes and consistency methods provide diagnostic tools (Alain and Bengio, 2017; Khanmohammadi et al., 2025). Semantic entropy measures uncertainty over answer meanings (Farquhar et al., 2024), a different object from our readout-subspace shifts. Activation patching and circuit analysis (Meng et al., 2022; Vig et al., 2020; Elhage et al., 2021; Olsson et al., 2022; Conmy et al., 2023; Yeo et al., 2025) require attention to corruption, metrics, and alignment (Zhang and Nanda, 2024; Heimersheim and Nanda, 2024). We retain these analyses as supporting diagnostics in Appendices D and E. Architectural equivariance (Cohen and Welling, 2016; Li et al., 2025) builds symmetry into models; EquiBench evaluates program equivalence (Wei et al., 2025). ORBIT-SIT evaluates actions induced by prompts without architectural changes, and OrbitAlign uses LoRA/QLoRA (Hu et al., 2022; Dettmers et al., 2023) to train on action-consistent targets.

Symbol	Meaning in the running example
T, P	Review text and clean sentiment instruction
$\omega(P)$	Instruction requesting the opposite label
$c_{\tau,0}$, $c_{\tau,\omega}$	Parsers into clean and rewritten output spaces
z, z_ω	Clean and rewritten canonical predictions
$z^\star$	Gold sentiment label
$g_{\tau,\omega}$	Declared label swap, $z \mapsto 1 - z$
$R^\star$	Error against the transformed gold answer
R°	Inconsistency with the transformed clean prediction

Table 2: Core symbols. The task index τ selects the dataset's canonical output space; a canonical answer is a deterministically parsed label or structure.

3 ORBIT-SIT: Framework

3.1 From the example to typed actions

In the introductory example, the task text is the film review, the clean instruction requests sentiment, and the rewrite requests a label swap. Table 2 connects these objects to the notation. The task text is fixed throughout a comparison.

Let $f_\theta : \mathcal{P} \times \mathcal{T} \to \mathcal{U}$ map prompts and texts to raw outputs. For task τ, a clean canonicalizer $c_{\tau,0} : \mathcal{U} \to \mathcal{Z}_{\tau,0}$ yields $z = c_{\tau,0}(f_\theta(P, T))$. Gold answers are $z^\star_\tau(T) \in \mathcal{Z}_{\tau,0}$. Each canonicalizer may also return a failure marker $\perp$ outside its valid answer space. Any comparison involving $\perp$ receives loss 1; the action is never applied to an invalid answer. In particular, two parse failures do not count as law compliance.

3.2 Prompt semantic operators

An operator $\omega : \mathcal{P} \to \mathcal{P}$ supplies a rewritten canonicalizer $c_{\tau,\omega} : \mathcal{U} \to \mathcal{Z}_{\tau,\omega}$ and a declared action $g_{\tau,\omega} : \mathcal{Z}_{\tau,0} \to \mathcal{Z}_{\tau,\omega}$. Write $z_\omega = c_{\tau,\omega}(f_\theta(\omega(P), T))$. Invariance operators preserve the space and use the identity action; the remaining operators, called equivariance operators here, apply a non-identity action. This usage includes type-changing maps and does not require a group action or an invertible transformation.

Binary flips are involutions. Hierarchy and projection are idempotent only after extension to act as the identity on already projected outputs. Existential queries are monotone under set inclusion. Appendix B states the domains and proofs; these are properties of the declared maps, not assumptions that models obey them.

3.3 The operator law and its scope

The model obeys the operator law for an item exactly when

$$z_\omega = g_{\tau,\omega}(z). \tag{1}$$

For a positive clean prediction under polarity, this requires a negative rewritten prediction. It says nothing by itself about whether the clean prediction is correct. Thus a model can obey the law while remaining task-incorrect. We measure both reference states below. Refusal, parse failure, and an ordinary-answer echo can all violate the contract; the violation alone does not identify its cause.

Why invariance is insufficient. Zero self-grounded risk on all invariance and equivariance operators implies stability under every tested preserving rewrite. The converse fails: a model that always emits the same binary label is perfectly stable but cannot obey a label flip. This establishes a strict separation between the full operator criterion and invariance-only stability (Appendix B.5). Gold-grounded risk is additionally needed to assess correctness.

3.4 Dual-risk evaluation

For a task- and operator-specific loss $d_{\tau,\omega}$, we define

$$R^\star_{\tau,\omega}(\theta) = \mathbb{E}\big[d_{\tau,\omega}(z_\omega, g_{\tau,\omega}(z^\star_\tau(T)))\big], \tag{2}$$
$$R^\circ_{\tau,\omega}(\theta) = \mathbb{E}\big[d_{\tau,\omega}(z_\omega, g_{\tau,\omega}(z))\big]. \tag{3}$$

The expectation is over evaluated text–prompt pairs. Binary tasks use 0/1 loss; structured tasks use their stated set or field-overlap loss. Parse failures receive maximal loss and are separately reported. Appendix A gives canonicalization examples. Low $R^\star$ means correctness against transformed gold; low R° means consistency with the transformed clean prediction. When the clean prediction equals gold, the two targets coincide. When it does not, these risks answer different questions.

3.5 Aggregate scores and diagnosis

Let Ω^{inv}_τ and Ω^{eqv}_τ be the disjoint sets of applicable invariance and equivariance operators. For nonempty families we report

$$\text{InvScore}_\tau = 1 - \frac{\sum_{\omega\in\Omega^{\text{inv}}_\tau} R^\star_{\tau,\omega}}{|\Omega^{\text{inv}}_\tau|}, \tag{4}$$
$$\text{EqvScore}_\tau = 1 - \frac{\sum_{\omega\in\Omega^{\text{eqv}}_\tau} R^\star_{\tau,\omega}}{|\Omega^{\text{eqv}}_\tau|}, \tag{5}$$
$$\text{OrbitConsistency}_\tau = 1 - \frac{\sum_{\omega\in\Omega_\tau} R^\circ_{\tau,\omega}}{|\Omega_\tau|}. \tag{6}$$

Higher scores are better. The $(1 - R^\star, 1 - R^\circ)$ plane separates gold-correct and orbit-consistent behavior; orbit-consistent but task-incorrect behavior; gold-hit but orbit-inconsistent behavior; and task-and-orbit non-compliance. These are descriptions of measured behavior, not labels for a model's intelligence. The midlines in Figure 2 are visual guides, not validated thresholds.

4 Eight Instantiated Operators

We instantiate three invariance operators (PARAPHRASE, REORDER, ALIAS) and five equivariance operators (POLARITY, COMPLEMENT, EXISTS, HIERARCHY, PROJECTION). For exposition we group them by dominant empirical failure mode—*surface-invariance*, *instruction-conflict* (POLARITY/COMPLEMENT), and *task-space morphism* (EXISTS/HIERARCHY/PROJECTION)—noting these are illustrative rather than strictly orthogonal categories: the operator law itself is defined uniformly via $g_{\tau,\omega}$. The invariance family leaves the canonical target unchanged, while the equivariance family applies a known semantic action: binary flips for SST-2 polarity and BoolQ complement, existential collapse for GoEmotions, fine-to-coarse projection for Few-NERD, and event-type projection for MAVEN.

Each operator is specified by three falsifiable ingredients: a prompt rewrite that induces ω in natural language, an operator-specific canonicalizer $c_{\tau,\omega}$, and an explicit action $g_{\tau,\omega}$ that rewrites the gold canonical state $z^\star$ into the target image. Evaluation therefore compares the model's canonicalized output to the *operator's image of the gold*, not to a heuristic paraphrase score or an LLM judge—a narrower but auditable test of operator-specified rewrites on top of the base task.

A worked example. For SST-2 polarity, the rewritten prompt asks the model to return the *opposite* label for the same sentence, and the semantic action sends every $z^\star \in \{0, 1\}$ to $1 - z^\star$. A

law-obeying model must therefore flip its own prediction on every item: remaining stable under the rewritten prompt is *by construction* a violation of Eq. (1), not a form of robustness. Because the canonical space is still binary for both the clean and polarity prompts, $R^\star$ and R° are directly comparable in this example. For EXISTS on GoEmotions the action is $g_{\exists,S}(z) = \mathbf{1}[z \cap S \neq \varnothing]$: the prompt asks whether at least one emotion from a candidate set S appears, and $c_{\tau,\omega}$ projects the perturbed output onto $\{0, 1\}$ before the risk is computed (the full 28-way emotion prediction remains the ordinary canonical space for invariance operators on the same dataset). This probes a failure mode paraphrase-only tests cannot observe, because paraphrasing never asks the task state to change type.

4.1 Algorithmic operator validation

Before any model is evaluated, the operator library itself is validated by a deterministic rule-based suite covering three levels: (i) surface constraints that the generated prompt must satisfy by construction (e.g., the HIERARCHY prompt never mentions a fine label), (ii) consistency of $g_{\tau,\omega}$ as a well-defined function on the gold canonical space, and (iii) implemented algebraic closure checks. Every instantiated operator passes the implemented suite on more than 500 synthetic realizations, with no reported surface, action, or closure failures (Appendix C.1). These tests do not exhaustively check every property proved in Appendix B. This makes the operator library auditable, while leaving natural-language construct validity to the separate semantic audit. A further *semantic gap* result shows that paraphrase robustness is strictly weaker than operator-law competence (Appendix B).

Human semantic audit. The deterministic suite verifies the formal contract; construct validity also requires the natural-language rewrite to be read as the intended action. We therefore run a human-facing audit as a separate screen: three annotators (blind to model outputs and to each other's labels) see the base prompt, the rewritten prompt, the task description, and the proposed action $g_{\tau,\omega}$, and judge whether the rewrite (i) preserves the target, (ii) applies the stated action, or (iii) introduces additional requirements. Across N=200 sampled realizations (25 per operator), the overall Fleiss' κ=0.83 indicates strong agreement; the binary operators (POLARITY, COMPLEMENT) reach near-perfect consensus ($\kappa \geq 0.90$), while the structured operators (EXISTS, HIERARCHY, PROJECTION) show agreement (κ=0.74–0.79), indicating that agreement varies across the scoring-space transformations. Fifteen prompts (7.5%) flagged as introducing additional requirements or misapplying the action were discarded and replaced before any model was scored. The audit therefore validates retained prompt realizations; all model scores remain computed by the canonical operator law above.

5 Controlled Behavioral Evidence

5.1 Experimental setup and interpretation

We evaluate Qwen2.5-7B-Instruct (Qwen Team, 2024), Qwen3-4B-Instruct-2507 (Qwen Team, 2025), and Phi-4-mini-instruct (Microsoft, 2025) on SST-2 (Socher et al., 2013), BoolQ (Clark et al., 2019), GoEmotions (Demszky et al., 2020), Few-NERD (Ding et al., 2021), and MAVEN (Wang et al., 2020). Open models use greedy decoding and four-bit NormalFloat (NF4) loading. Each dataset has three invariance operators and one applicable equivariance operator, giving 20 model–dataset–operator cells per backbone.

We distinguish the behavior evaluation, full repair evaluation, matched ablation, and wording-transfer pools. *Adapter-disjoint* means zero example-ID overlap between adapter training and evaluation. *Held-out wordings* are previously unseen natural-language realizations of the same action, not unseen operator families. Table 6 identifies the two repair pools most important for interpreting the headline result.

The reported confidence intervals (CIs) use percentile bootstrap resampling (Efron and Tibshirani, 1994); the experimental protocol uses 10,000 replicates and paired comparisons. Where multiple-testing results are reported, q denotes a Benjamini–Hochberg adjusted p-value (Benjamini and Hochberg, 1995). Controls with multiple wordings use item-by-wording resampling. We distinguish observed changes from tests of significance and retain the available paired CIs rather than inferring uncertainty from rounded aggregate scores.

When a conditional estimate is informative. Self-grounded risk remains defined even when the base answer is wrong. However, attributing a violation specifically to executing an action requires a sufficiently large correct-base subset. Our binary-control screen requires parse success at least 98%, clean accuracy at least 75%, at least 200 correct-base observations, and CI half-width no greater

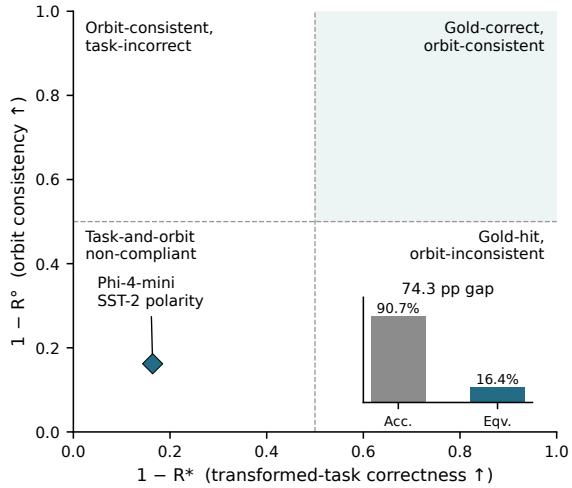


Figure 2: Two distinct diagnostic axes (higher is better). The point shows the reported Phi-4-mini SST-2 polarity cell: $(1 - R^{\star}, 1 - R^{\circ}) = (0.164, 0.162)$. Quadrant names describe behavior; the 0.5 midlines are visual guides, not capability thresholds. The inset compares clean accuracy with EqvScore on this cell. Full per-operator gold-grounded results appear in Figure 3.

than 7.5 percentage points. These criteria screen the conditional analysis, not the unconditional risk definition. Structured cells that fail them are diagnostics only. For example, the MAVEN projection conditional observation is 0 violations among 4 correct-base items, which is too small to establish reliable execution.

5.2 Accuracy and operator compliance diverge

Phi-4-mini achieves 90.7% clean SST-2 accuracy but 16.4% EqvScore under polarity, a 74.3-point gap. Its corresponding $(R^{\star}, R^{\circ}) = (0.836, 0.838)$ indicates both gold-grounded error and inconsistency with the requested action on its clean prediction (Figure 2). Qwen2.5-7B has much lower polarity risk ($R^{\star} = 0.055$), whereas BoolQ complement risks range from 0.298 to 0.642 across the three models. Thus the difficulty is model- and operator-dependent, rather than a universal inability to flip a label.

Accuracy-oriented baselines. Table 3 compares prompt averaging and paraphrasing with the gold-grounded invariance score. Their binary-task values are similar, as expected when the requested action is identity. This does not make them tests of an answer-changing instruction.

Dataset	Single	Multi	Dyn.	InvScore
SST-2	.929	.933	.934	.935
BoolQ	.833	.834	.835	.835
Few-NERD	.449	.457	.461	.459
GoEmotions	.192	.167	.160	.159
MAVEN	.018	.019	.019	.020

Table 3: Qwen2.5-7B behavior-pool scores (higher is better): single prompt, multi-prompt average, dynamic paraphrasing, and InvScore. Structured-task scores use their task loss and are not binary exact-match accuracy. They should not be compared directly with the strict clean-accuracy checks after adaptation.

Quantity	Estimate
Ordinary accuracy change in dual-field output	-1.1 pp
Codebook gap from clean accuracy	17.8 pp
95% CI	$[12.4, 23.1]$
Law failure given ordinary answer correct	14.6%
95% CI	$[9.8, 19.7]$
Explicit refusal among violations	2.8%
Ordinary-answer echo among violations	67%
Operator-only / joint dual-field compliance	97.4/72.1%

Table 4: Non-deceptive binary controls: 400 items per task–model cell, five wordings; pp means percentage points. Conditional rates use the ordinary-correct subset; refusal and echo use the violation subset. The ordinary-accuracy change ranges from -0.6 to -1.8 points. The reported intervals resample items and wordings.

Parsing and base-task difficulty. Across structured model–operator cells, conditioning on successful parsing changes mean scores by at most 0.027 (Table 9). This shows that parsing alone does not explain the low scores; it does not show that all remaining errors are operator failures. In particular, sparse correct-base subsets prevent precise action-specific estimates on MAVEN. The binary controls below provide the stronger interpretive evidence. The API snapshot in Appendix C.7 is additional behavioral evidence, not a controlled scaling study.

5.3 Non-deceptive controls

We test SST-2 polarity and BoolQ complement on Phi-4-mini and Qwen2.5-7B using 400 items per cell and five wordings. Three controls separate the output contract from asserting something believed false: (i) neutral A/B or X/Y codebooks, explicitly defined as codes; (ii) two fields, `ordinary_answer` and `operator_output`; and (iii) an operator-only oracle supplied with the clean label. Table 4 reports the completed control results.

Ordinary accuracy changes little, yet a 17.8-point residual remains under neutral codebooks.

Condition	LLM calls	Compliance
Direct instruction	1	38.6%
Explicit two-step instruction	1	61.2%
Infer, then transform	2	96.8%
Deterministic postprocessor	1 + rule	100%
OrbitAlign, held-out wording	1	91.4%

Table 5: Transformation-compliance controls (higher is better) on the binary-control setting, 400 items per cell and five wordings. Executing a mapping correctly does not ensure that its input label was gold-correct; the deterministic result is not 100% task accuracy. OrbitAlign changes model parameters and uses unseen wordings.

Explicit refusal accounts for 2.8% of violations, whereas copying the ordinary answer accounts for 67%. The operator-only control exceeds joint dual-field compliance by 25.3 points. These observations support a difficulty in binding inference to the requested mapping under this interface. They do not establish that every violation has one cause, or that refusal is undesirable.

5.4 Separating inference from transformation

On the same binary-control cells, an explicit one-call two-step prompt first asks for the ordinary answer and then its transformation. It improves transformed compliance from 38.6% to 61.2%, a 22.6-point gain, but retains an 18.1-point gap from clean accuracy (95% CI $[13.0, 23.4]$). A two-call infer-then-transform pipeline reaches 96.8%, and a deterministic postprocessor applies the binary mapping exactly (100%). OrbitAlign reaches 91.4% in one call on held-out wordings (Table 5).

This comparison makes the audit actionable: test prompt clarification and decomposition first; use an external operator when the interface permits it; consider adaptation when a single-call interface is required. The two-call baseline outperforms the reported one-call adapter result, so these interventions occupy different cost and interface regimes rather than establishing an unconditional ranking.

6 OrbitAlign: Repair and Transfer

OrbitAlign trains on declared action targets using QLoRA (Dettmers et al., 2023), without updating all backbone parameters. Each base example $(P, T, z^\star)$ supplies an ordinary task row, three invariance rows with target $z^\star$, and an applicable equivariance row with target $g_{\tau,\omega}(z^\star)$. The objective is

$$\mathcal{L} = \lambda_{\text{task}}\mathcal{L}_{\text{task}} + \lambda_{\text{inv}}\mathcal{L}_{\text{inv}} + \lambda_{\text{eqv}}\mathcal{L}_{\text{eqv}}. \quad (7)$$

BoolQ repair pool	n	Base	Orbit	Reduction
Full evaluation	400	.441	.187	.254
Matched ablation	200	.298	.105	.193

Table 6: Pool-specific gold-grounded risks (lower is better) for Qwen2.5-7B complement. Reduction is Base minus OrbitAlign on the same items; n is the reported evaluation-pool size. The source reports aggregate risks, not integer error counts; its aggregation over wordings or runs is not fully specified.

Each term is mean token-level cross-entropy for its family within a partitioned micro-batch. Thus the objective supervises transformed gold outputs; it does not directly minimize the self-grounded risk. We use $\lambda_{\text{task}} = 1$ and $\lambda_{\text{inv}} = \lambda_{\text{eqv}} = 0.5$, LoRA rank 16 and scaling 32, 400 base items per dataset, and one epoch. AdamW uses learning rate 2×10^{-4}, batch size 2, and gradient accumulation 8. Training and evaluation share no item identifiers.

6.1 Repair results and matched ablation

On the full 400-item Qwen2.5-7B repair pool, BoolQ complement risk falls from 0.441 to 0.187: a paired reduction of 0.254 ($q < 10^{-3}$). The 200-item matched four-condition ablation is a different pool, with risks 0.298 and 0.105 and reduction 0.193. Table 6 reports each calculation explicitly. Neither delta is obtained by subtracting across pools.

The matched ablation compares task-only adaptation, uniform cross-entropy on all operator-expanded rows, and family-partitioned OrbitAlign (Table 10). Task-only training does not reliably repair the binary flips. Operator augmentation closes most of their gap: uniform training ties OrbitAlign on SST-2 and slightly improves on it for BoolQ (0.100 versus 0.105). Partitioning adds larger gains on structured tasks, reducing MAVEN projection risk by a further 4.4 points and Few-NERD hierarchy by 1.7 points relative to uniform training. This is evidence for the particular objective comparison, not proof of unrestricted operator generalization.

6.2 Wordings and backbone replication

Unseen-wording evaluation uses 200 adapter-disjoint items per binary task. Qwen2.5-7B SST-2 risk changes from 0.070 to 0.045, and BoolQ complement from 0.785 to 0.190 (Table 12). The action stays fixed while its natural-language realization changes.

Applying the same training recipe to Qwen3-

4B and Phi-4-mini lowers gold-grounded risk in 20/20 and 19/20 cells, respectively (Table 14). The exception is Phi-4-mini on GoEmotions EXISTS, which increases by 0.015; we retain that regression. Across all three backbones, the completed replication summary reports 59/60 non-increasing cells. Non-increase includes ties and is not a statement of statistical significance in every cell. Held-out clean accuracy increases on every dataset in the within-task repair checks (Tables 11 and 15). Replicating a fixed recipe on a new backbone requires training an adapter on that backbone; it is not zero-shot transfer of one adapter's weights.

6.3 Same-action cross-task transfer

We also train on one binary task and evaluate the shared action $z \mapsto 1 - z$ on the other task using held-out items and wordings. SST-2 polarity to BoolQ complement reduces $R^\star$ from 0.482 to 0.331; the reverse transfer reduces it from 0.518 to 0.389 (Table 7). Matched task-only and token-matched augmentation controls have smaller risk changes, approximately -0.01 and -0.02, respectively. Leave-one-out and composition checks improve 11/14 cells, with three seeds agreeing and reported standard deviation below 0.03. These results support transfer of the tested binary action across tasks and wordings. They do not establish generalization to arbitrary unseen action families.

7 Discussion and Conclusion

What the audit establishes. ORBIT-SIT evaluates a declared task-space action using two reference states. Gold-grounded risk asks whether the transformed answer is correct; self-grounded risk asks whether it follows the transformation of the model's clean answer. This distinction exposes a 74.3-point accuracy–EqvScore gap on Phi-4-mini SST-2 and avoids rewarding consistently incorrect outputs. The central contribution is an auditable evaluation protocol with controlled interpretation.

What the controls change. Neutral codebooks and separate answer fields reduce the honesty confound but leave measurable non-compliance. The stronger two-call baseline shows that isolated transformation and joint execution are different demands. Refusal can be appropriate policy behavior; our metrics assess the stated contract and do not rank intelligence or truthfulness. When postprocessing is available, the declared action may be better enforced outside the model.

Diagnosis and repair. Semantic Leakage Ratio (SLR) and Action Consistency Index (ACI) describe readout-subspace shifts in supporting analyses (Appendix D). Alignment-aware patching provides little recovery in the tested settings and does not support single-layer localization (Appendix E). OrbitAlign instead changes distributed parameters through low-rank adapters. Its gains extend to unseen wordings, two further backbones, and same-action cross-task transfer, while its limitations remain visible in the retained regressions and untested action families.

Outlook. Reliable instruction interfaces need tests of both task correctness and requested transformations. The binary tasks offer interpretable evidence; structured tasks demonstrate wider scoring spaces while exposing limits of conditional estimation. Future work can test new action families, languages, and multi-step interfaces using the same separation between a declared contract, its measurement, and the scope of the conclusions.

Limitations

ORBIT-SIT measures compliance with a declared task-space action. It is not a sufficient test of general understanding, truthfulness, or safe behavior. In particular, a model may appropriately refuse an instruction; a contract violation must be interpreted alongside the refusal and non-deceptive controls. The deterministic suite validates the stated maps and surface constraints, and the 200-prompt human audit checks readability, but neither establishes validity for every natural-language realization.

The experiments use English benchmarks and a small number of model families. Pretraining contamination cannot be excluded. Binary tasks provide the primary evidence because the action and output space are clear; structured tasks often have too few correct-base examples for precise conditional estimates. Parse-conditioned scores alone cannot separate task error from action error. Some result summaries lack exact conditional counts, complete CIs, per-item predictions, or full environment manifests, which limits independent statistical reconstruction. The supplied reference implementation has incomplete structured-output parsers and an outdated GPU dependency specification; it does not establish the parser or environment used for every reported run (Appendix A). Reported aggregate controls should be interpreted within those bounds.

Training action → evaluation action	Base $R^\star$	Adapted $R^\star$	Paired $\Delta R^\star$ (95% CI)	Clean Δ (pp)
SST-2 polarity → BoolQ complement	.482	.331	$-.151$ $[-.194, -.107]$	-0.3
BoolQ complement → SST-2 polarity	.518	.389	$-.129$ $[-.171, -.088]$	-0.5

Table 7: Same-action transfer, with evaluation items and wordings held out. Lower risk is better. The small clean-accuracy decreases are retained; the absence of regression in within-task repair does not imply absence of regression in cross-task transfer.

OrbitAlign is supervised on the same declared laws that are audited. Disjoint items test item generalization; unseen wordings test new realizations of known actions. Same-action cross-task transfer and 14 leave-one-out/composition checks provide bounded additional evidence; they do not establish arbitrary cross-family generalization. We retain the Phi-4-mini GoEmotions regression and small clean-accuracy decreases under cross-task transfer. The two-call baseline can outperform the one-call adapter, and no exhaustive hyperparameter search is claimed. Representation and patching results are descriptive or specific to the tested interventions; absence of recovery at selected layers is not absence of a causal mechanism.

Ethical Considerations

Failure-mode audits can aid model evaluation but can also inform adversarial prompt design. The operators here perform benign task transformations; their scores should not be used to justify bypassing safety policies. We report aggregate behavior and do not include an automated adversarial-prompt search utility. Public benchmark text may contain personal names or offensive language. We did not collect new user data. The review supplement contains short benchmark-derived excerpts; inherited dataset risks and source access conditions still apply. Appendix A records the license limits, including unclear SST data-license metadata.

The semantic audit used three internal research collaborators, with no external crowdsourcing. They assessed prompt–action readability without model outputs or each other's labels. Recruitment beyond this internal role, separate payment, and demographic details are not reported. The authors' institutional-policy declaration states that ethics-board review was not required because the audit involved internal collaborators and collected no personal or sensitive data. This is not a claim of ethics-board approval or a formal exemption decision. Original dataset publications document their collection procedures.

Acknowledgments

We thank the reviewers for suggestions that improved the controls, reporting, and exposition. ChatGPT and GitHub Copilot assisted with prose and code suggestions during the original work. During camera-ready preparation, OpenAI Codex assisted with substantive editorial restructuring, integration of author-reported results, mathematical and numerical consistency checks, bibliographic checking, and preparation of explanatory figures and submission materials. The experimental evidence is the authors' reported work; AI-generated text is not independent experimental evidence. The human authors are responsible for verifying the final content, citations, and figures and for the decision to submit.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations Workshop*.

Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multilinguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Steven Cho, Stefano Ruberto, and Valerio Terragni. 2025. Metamorphic testing of large language models for natural language processing. In *Proceedings of the 2025 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 174–186.

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.

Taco Cohen and Max Welling. 2016. Group equivariant convolutional networks. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 2990–2999, New York, New York, USA. PMLR.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, pages 16318–16352. Curran Associates, Inc.

I. de Zarzà, J. de Curtò, Jordi Cabot, Pietro Manzoni, and Carlos T. Calafate. 2026. Semantic invariance in agentic AI. *Preprint*, arXiv:2603.13173.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Hai-Tao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. 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 3198–3213, Online. Association for Computational Linguistics.

Bradley Efron and Robert J. Tibshirani. 1994. *An Introduction to the Bootstrap*. Chapman and Hall/CRC.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Naïm Es-sebbani, Esteban Marquer, Yakoub Salhi, and Zied Bouraoui. 2026. Evaluating robustness of reasoning models on parameterized logical problems. *Preprint*, arXiv:2602.12665.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. 2017. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1):12140.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Manuel Frank and Haithem Afli. 2026. PTEB: Towards robust text embedding evaluation via stochastic paraphrasing at evaluation time with LLMs. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2832–2851, Rabat, Morocco. Association for Computational Linguistics.

Eliya Habba, Ofir Arviv, Itay Itzhak, Yotam Perlitz, Elron Bandel, Leshem Choshen, Michal Shmueli-Scheuer, and Gabriel Stanovsky. 2025. DOVE: A large-scale multi-dimensional predictions dataset towards meaningful LLM evaluation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11744–11763, Vienna, Austria. Association for Computational Linguistics.

Stefan Heimersheim and Neel Nanda. 2024. How to use and interpret activation patching. *Preprint*, arXiv:2404.15255.

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*.

Cheongwoong Kang, Jongeun Baek, Yeonjea Kim, and Jaesik Choi. 2025. When format changes meaning: Investigating semantic inconsistency of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2647–2667, Suzhou, China. Association for Computational Linguistics.

Reza Khanmohammadi, Erfan Miahi, Mehrsa Mardikoraem, Simerjot Kaur, Ivan Brugere, Charese H. Smiley, Kundan Thind, and Mohammad M. Ghassemi. 2025. Calibrating LLM confidence by probing perturbed representation stability. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10448–10514, Suzhou, China. Association for Computational Linguistics.

Zongzhao Li, Jiacheng Cen, Bing Su, Tingyang Xu, Yu Rong, Deli Zhao, and Wenbing Huang. 2025. Large language-geometry model: When LLM meets equivariance. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 34054–34073. PMLR.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Microsoft. 2025. Phi-4-Mini technical report: Compact yet powerful multimodal language models via mixture-of-LoRAs. *Preprint*, arXiv:2503.01743.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

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. *Preprint*, arXiv:2209.11895.

Qwen Team. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Qwen Team. 2025. Qwen3-4B-Instruct-2507 model card. Model card. Accessed September 15, 2026.

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.

Mikhail Seleznyov, Mikhail Chaichuk, Gleb Ershov, Alexander Panchenko, Elena Tutubalina, and Oleg Somov. 2025. When punctuation matters: A large-scale comparison of prompt robustness methods for LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20370–20385, Suzhou, China. Association for Computational Linguistics.

Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson. 2007. Topological methods for the analysis of high dimensional data sets and 3D object recognition. In *Eurographics Symposium on Point-Based Graphics*, pages 91–100. The Eurographics Association.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A massive general domain event detection dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.

Anjiang Wei, Jiannan Cao, Ran Li, Hongyu Chen, Yuhui Zhang, Ziheng Wang, Yuan Liu, Thiago S. F. X. Teixeira, Diyi Yang, Ke Wang, and Alex Aiken. 2025. EquiBench: Benchmarking large language models' reasoning about program semantics via equivalence checking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33868–33881, Suzhou, China. Association for Computational Linguistics.

Wei Jie Yeo, Ranjan Satapathy, and Erik Cambria. 2025. Towards faithful natural language explanations: A study using activation patching in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10425–10447, Suzhou, China. Association for Computational Linguistics.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, pages 1651–1678.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Yujia Zheng, Tianhao Li, Haotian Huang, Tianyu Zeng, Jingyu Lu, Chuangxin Chu, Yuekai Huang, Ziyou Jiang, Qian Xiong, Yuyao Ge, and Mingyang Li. 2026. Are all prompt components value-neutral? understanding the heterogeneous adversarial robustness of dissected prompt in large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7991–8019, Rabat, Morocco. Association for Computational Linguistics.

Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. 2024. PromptBench: A unified library for evaluation of large language models. *Journal of Machine Learning Research*, 25(254):1–22.

Mikel Zubillaga, Naiara Perez, Oscar Sainz, and German Rigau. 2026. SemBench: A universal semantic framework for LLM evaluation. In *Proceedings of*

the Fifteenth Language Resources and Evaluation Conference, pages 4597–4610, Palma de Mallorca, Spain. ELRA Language Resource Association.

A Reproducibility checklist

This appendix documents the experimental protocol and resources that support the results reported in the main paper. The main results use English benchmarks; the binary tasks are sentiment classification and passage-grounded yes/no question answering, and the structured tasks cover emotion sets, entity typing, and event detection.

Computational budget. The reported local computational budget is approximately 30–50 GPU-hours. Experiments were conducted on a workstation equipped with two 24 GB-class consumer GPUs under 4-bit NF4 quantization. Measured wall-clock times were: roughly 8–14 hours for the behavior layer (15 model $\times$ dataset cells), 2–3 hours for the representation and topology probes, 6–10 hours for patching controls (11 operator targets, three alignment strategies, three decoder layers each), and under 30 minutes for one OrbitAlign QLoRA pass on a single 24 GB GPU. The held-out wording check in Appendix C.5 uses two parallel 4-bit Qwen2.5-7B inference jobs on the two GPUs and completes in under two minutes for the 400 total examples. Auxiliary analysis and figure generation together occupy under 30 minutes of CPU time; the parse-conditioned structured-task analysis in Appendix C.2 is CPU-only and reads existing behavior summaries without additional model inference.

Determinism. Every experiment fixes a global random seed before model loading and uses greedy decoding; device placement is pinned so representation probes and causal interventions attach to identical modules across runs. The reported procedure fixes resampling seeds and uses 10 000 bootstrap replicates. The supplied reference configurations and data card specify seed 42, greedy decoding, a 64-token generation cap (256 for MAVEN), and float16 NF4 computation. The three-seed transfer summary does not identify the other seeds. A pinned requirements file is included, but no execution manifest establishes the versions used for each historical run. Its Transformers 4.40.2 pin is incompatible with Qwen3's documented minimum of 4.51.0 (Qwen Team, 2025), so it is not a usable three-backbone environment specification.

Data provenance. We use SST-2, BoolQ, GoEmotions, Few-NERD supervised, and MAVEN in their publicly released forms, each retrieved from the original distributor and cached locally for offline experimentation. We use these benchmarks for research evaluation. The review supplement contains short benchmark-derived excerpts in its schema demonstrations, but does not redistribute the complete datasets. Underlying source rights still apply to those excerpts. BoolQ data are CC BY-SA 3.0; Few-NERD data are CC BY-SA 4.0 (its code is separately Apache-2.0); GoEmotions' source repository uses Apache-2.0; and the MAVEN repository uses MIT. The Stanford SST release does not provide a clear standalone data-license statement; we do not infer unrestricted redistribution rights for its movie-review text. Source dataset publications and distributors remain authoritative for the original collection and access conditions.

Model provenance. Qwen2.5-7B-Instruct and Qwen3-4B-Instruct-2507 are released under Apache-2.0; Phi-4-mini is released under MIT. Auxiliary embedding encoders (Qwen3-Embedding-0.6B under Apache-2.0 and BGE-M3 under MIT) are used only for diagnostic computations (Zhang et al., 2025; Chen et al., 2024). All weights are obtained from their official distributors and used fully offline.

Statistical reporting. The experiment protocol uses 10,000 percentile-bootstrap replicates and paired comparisons. Reported intervals and adjusted tests are identified alongside the corresponding results; point estimates are not presented as confidence intervals. Multiple-testing correction uses the Benjamini–Hochberg procedure. Rebuttal controls resample items and wordings, and the additional transfer/composition checks report three seeds with standard deviation below 0.03. Per-item denominators and complete uncertainty summaries are not recoverable from all aggregate result records, a reporting limitation made explicit in Appendix C.

Algorithmic operator validation. Operator semantics are validated by a deterministic rule-based suite (Appendix B) that checks surface constraints, g-action consistency on gold targets, and the algebraic closure properties of §3.2 on a fixed bank of synthetic inputs per operator. All instantiated operators pass the implemented checks. This does

not mean that the reference validation CLI exhaustively checks every stated property: for example, it does not test existential monotonicity. The algebraic statements themselves are proved in Appendix B. This protocol targets the algebraic and syntactic contracts that the operator library is required to respect, and does not purport to subsume task-specific human semantic judgment.

Evaluation pools and disjointness. OrbitAlign is trained and evaluated on disjoint example pools: the adapter training pool and the held-out evaluation pool share no example identifier, enforced at sampling time. Behavior-layer and OrbitAlign evaluation use independent held-out subsamples, so the absolute risks reported in §5 and §6 are defined on different pools and are to be read as domain-consistent rather than point-identical. The full BoolQ repair pool has 400 evaluation items and the matched four-condition ablation has 200; unseen-wording checks use 200 per binary task. Available paired statistics are reported with their corresponding tables; the pools must not be interchanged.

Responsible compute. We report approximate GPU-hour cost per stage and reuse cached clean-prompt hidden states between the behavior, representation, geometry, topology, and patching stages to avoid redundant forward passes. The representation and patching analyses operate on a fixed-size per-dataset subsample to contain compute without biasing the comparison.

Reproducibility and artifact scope. The supplied Software/Data supplements contain a CPU reference implementation, configurations, prompt templates, alias resources, and records explicitly marked as schema demonstrations. There are 30 evaluation-record rows across three files (including five paired before/after examples), 40 human-audit rows, and partial train/evaluation ID manifests. These demonstrations are not random samples for re-estimating the paper's results. The declared behavior pool is 200 items per cell and the training pool is 400 items per dataset; the later full-repair and control pools are identified separately above. The supplied GPU/API entry points only load configuration files and print dry-run summaries; they do not implement end-to-end inference or adapter training. Full evaluation logs, adapter weights, the complete human-audit ledger, and representation/patching outputs are not included. The archived human-audit excerpt contains two annotations outside the documented three-category scheme; the full adjudication log is unavailable. The reported $\kappa = 0.83$ and 185/200 pass count therefore cannot be reconstructed from that excerpt. Consequently, the supplements support inspection of selected framework components, not independent reconstruction of every empirical result. We do not claim an available permanent full-artifact release.

Reference implementation boundaries. The supplied code passes its 30 CPU unit tests, but those tests do not cover the complete experimental pipeline. In particular, the MAVEN type-only reference parser still expects trigger–type pairs, and the Few-NERD reference parser does not enforce the complete official type vocabulary. Its coarse mode explicitly recognizes only four bare categories; its tokenizer also does not implement the multiword aliases in the accompanying alias file. These paths require completion before the reference code can implement the structured-task protocol below. The mathematical operator definitions and author-reported experimental results are distinguished from this incomplete reference implementation; the archive does not establish which parser revision produced each historical aggregate. Structured-task findings retain their diagnostic scope. Dataset text and model weights should be obtained from their original distributors under the applicable terms.

Canonicalization and distance examples. To clarify the measurement pipeline, we provide concrete examples for each task family:

- **SST-2 / BoolQ (binary tasks).** The canonicalizer extracts the first occurrence of a target label token ("positive"/"negative" for SST-2; "true"/"false" for BoolQ) from the model's raw text output, case-insensitively, and maps to $\{0, 1\}$. If no label token is found, the item is marked as a parse failure and scored conservatively (d=1). The distance is 0/1 loss on the binary label. The alias operator instead uses a synonym-aware binary canonicalizer mapped to the same two labels.
- **GoEmotions (multi-label emotion set).** The canonicalizer parses a comma-separated list of emotion names from the model output and normalizes each name to the official 28-class taxonomy via a hand-written alias table (e.g. "happy"→"joy", "scared"→"fear"). Unrecog-

nized tokens are discarded; if no recognized emotion remains, the item is a parse failure. The clean/invariance distance is $1-\text{Jaccard}(A, B)$ between emotion sets. Under EXISTS, the rewritten output is binary and is compared with $\mathbf{1}[z \cap S \neq \varnothing]$ using 0/1 loss, for the declared candidate set S.

- **Few-NERD (entity typing).** Each entity mention is canonicalized to its *fine-grained type* string (e.g. "person-athlete", "location-GPE"). The documented protocol normalizes schema variants (e.g. "sports player" $\rightarrow$ "person-athlete") before comparison; the archive's alias file lists 20 entries, but its parser does not apply all of them. Under hierarchy, the rewritten canonicalizer should instead return coarse parent types. The distance between two entity-type sets is $1-$Jaccard overlap on the type strings, computed after alias normalization.
- **MAVEN (event detection).** The canonicalizer extracts `(trigger, event_type)` pairs from JSON-formatted output. Each pair is canonicalized by lowercasing the trigger span and matching the event type in the MAVEN taxonomy. The intended protocol checks the official vocabulary; the supplied reference parser does not perform that vocabulary check. Distance is $1 - \text{F}_1$ on the set of canonical pairs, using string equality on both trigger and type. Under projection, the required rewritten canonicalizer returns a type set, compared with the type projection of the corresponding reference. A failed required parse is scored as $d{=}1$.

In all cases, the canonicalizer is *deterministic* and does not call any LLM. Parse-failure rates are summarized in Appendix C; they are reported below 5% for Qwen2.5-7B.

Representative prompt rewrite examples. We give schematic examples for the equivariance operators on their target datasets. These clarify the contract; they do not replace the exact prompt banks and candidate-set specifications used in a run. The supplied GoEmotions template uses $S = \{\text{anger}, \text{fear}, \text{joy}, \text{sadness}\}$; this does not establish the candidate set for every historical run. Each example shows: (i) the base prompt P, (ii) the rewritten prompt $\omega(P)$, and (iii) the semantic action $g_{\tau,\omega}$ applied to the canonical output.

- **Polarity (SST-2).** P: "Classify the sentiment of the following review as positive or negative: `[T]`". $\omega(P)$: "Classify the sentiment of the following review, but return the *opposite* label (if the true sentiment is positive, output negative; if negative, output positive): `[T]`". g: $z \mapsto 1-z$ on $\{0, 1\}$.
- **Complement (BoolQ).** P: "Answer the following yes/no question with 'true' or 'false': `[T]`". $\omega(P)$: "Answer the following yes/no question, but return the complementary truth value (if the correct answer is true, output false; if false, output true): `[T]`". g: $z \mapsto 1-z$ on $\{0, 1\}$.
- **Exists (GoEmotions).** P: "List all emotions expressed in the following text (choose from the 28 GoEmotions labels): `[T]`". $\omega(P)$: "For the following text, determine whether *any* emotion from the specified candidate set `[S]` is expressed. Output 'true' if at least one candidate emotion is present, 'false' otherwise: `[T]`". $g_{\exists,S}$: $z \mapsto \mathbf{1}[z \cap S \neq \varnothing]$, where z is the emotion set and S is the declared (fixed) candidate-emotion set drawn from the GoEmotions taxonomy; the full 28-way set collapses onto $\{0, 1\}$.
- **Hierarchy (Few-NERD).** P: "Extract all named entities and assign each a fine-grained type from the Few-NERD schema (e.g. person-athlete, location-GPE): `[T]`". $\omega(P)$: "Extract all named entities but assign each only its *coarse-grained* type (the part before the hyphen, e.g. person, location, organization): `[T]`". g: π_{h}: maps each fine type to its coarse parent (person-athlete $\rightarrow$ person).
- **Projection (MAVEN).** P: "Detect all events in the following text and output each as a (trigger, event_type) pair using the MAVEN schema: `[T]`". $\omega(P)$: "For the following text, output only the set of *distinct event types* that are triggered (ignore trigger spans; output just the type labels): `[T]`". g: $\{(t_i, e_i)\} \mapsto \{e_i\}$; projects from trigger–type pairs onto the type set.

Canonicalizer input–output examples. Clean and rewritten outputs use deterministic canonicalizers matched to their declared types. They are the same for most preserving rewrites and binary flips (with synonym handling for alias), but differ for the type-changing operators. The following are intended protocol examples; the reference-code coverage limits above apply:

- SST-2: raw "The label is Positive." $\mapsto$ `positive`; under POLARITY, the target image is `negative`.

- BoolQ: raw "Answer: false" $\mapsto$ `false`; under COMPLEMENT, the target image is `true`.
- GoEmotions: raw "happy, a little scared" $\mapsto$ $\{\texttt{joy}, \texttt{fear}\}$ via aliases; under EXISTS, the action is true only when the recognized set intersects S.
- Few-NERD: raw "Jordan: sports player; Chicago: city" $\mapsto$ $\{\texttt{person-athlete}, \texttt{location-GPE}\}$ after alias normalization; under HIERARCHY, these map to $\{\texttt{person}, \texttt{location}\}$.
- MAVEN: raw JSON with trigger `resigned` and type `Resignation` $\mapsto$ $\{(\texttt{resigned}, \texttt{Resignation})\}$; under PROJECTION, this maps to $\{\texttt{Resignation}\}$.

B Algebraic Properties and the Stability Gap

These statements concern the declared semantic actions, not a claim that natural-language prompt composition automatically realizes the corresponding composed action. All equalities assume successful canonicalization into the stated spaces.

B.1 Binary flips are involutions

Proposition 1. *For polarity and complement, $g(z) = 1 - z$ on $\{0, 1\}$ satisfies $g \circ g = \mathrm{id}_{\{0,1\}}$.*

Proof. For either binary action, $g(g(z)) = 1-(1-z) = z$. □

B.2 Type-changing projections and idempotence

Hierarchy maps a set of fine-grained entity types X into a set of coarse types Y. Projection maps trigger–event-type pairs X into event-type sets Y. In either case, $g : X \to Y$ changes the type of the output. Thus $g \circ g$ is not defined on X without an extension; in particular, coarse label strings need not belong to the fine-label vocabulary.

Proposition 2. *On the tagged disjoint union $X \sqcup Y$, extend a projection by $\bar{g}(x) = g(x)$ for $x \in X$ and $\bar{g}(y) = y$ for $y \in Y$. Then $\bar{g} \circ \bar{g} = \bar{g}$.*

Proof. The first application always returns an element tagged as belonging to Y. The second application is the identity on that output. □

This is the precise sense in which dropping an already-dropped field or retaining an already-coarsened type is idempotent. It does not assert that two type-incompatible prompt rewrites can be composed.

B.3 Existential monotonicity

For a fixed candidate-label set $S \subseteq \mathcal{Y}$, define $g_{\exists,S}(z) = \mathbf{1}[z \cap S \neq \varnothing]$.

Proposition 3. *If $z \subseteq z'$, then $g_{\exists,S}(z) \leq g_{\exists,S}(z')$. If $S \neq \varnothing$ and $S \subseteq z$, then $g_{\exists,S}(z) = 1$.*

Proof. Every member of $z \cap S$ also belongs to $z' \cap S$, so a nonempty intersection cannot become empty when z grows. If $S \subseteq z$ and S is nonempty, then $z \cap S = S \neq \varnothing$. □

B.4 Invariance is the identity action

For $\omega \in \Omega_\tau^{\mathrm{inv}}$, the action is $g_{\tau,\omega} = \mathrm{id}_{\mathcal{Z}_{\tau,0}}$ by definition. The instantiated paraphrase, reorder and alias operators change the prompt realization while preserving its canonical task target.

B.5 Why stability alone is insufficient

Let *invariance stability* mean that the clean and rewritten canonical predictions agree for every invariance operator. Let *operator compliance* require the declared law for every instantiated operator, including nonidentity actions.

Proposition 4. *Operator compliance implies invariance stability. The converse fails for a library containing a binary flip.*

Proof. For an invariance operator, the law is exactly equality between the clean and rewritten answers because g is the identity. Conversely, consider a model returning the fixed binary label 0 under every prompt. It is perfectly stable under every invariance rewrite, but under a flip it still returns 0 although $g(0) = 1$. It therefore violates the operator law. □

Gold correctness is a separate requirement. A model can satisfy a self-grounded law while acting consistently on a wrong clean answer. For example, on an item with gold label 1, a model returning 0 under all invariance prompts has zero invariance self-risk and unit gold risk. Reporting both risks exposes this distinction.

The gold-grounded quantity $\mathrm{InvScore} = 1$ alone does not guarantee clean-to-rewrite stability: all rewritten predictions may be correct while the clean prediction is wrong. The implication does hold when clean correctness is additionally required. For

a general nonbinary distance, $1 - \mathbb{E}[d]$ is an expected similarity score; it is a probability of exact agreement only when d is zero–one loss.

C Additional Results and Controls

This appendix provides cell-level and layerwise diagnostics for readers who wish to inspect the evidence behind the behavior and mechanism sections in more detail.

C.1 Operator-contract validation

Table 8 reports the deterministic validation suite run before model evaluation. For each operator we sample more than 500 synthetic realizations across its applicable datasets and check three contracts: surface constraints of the prompt rewrite, consistency of the semantic action $g_{\tau,\omega}$ on gold canonical states, and the implemented algebraic closure checks. All operators pass those checks. The reference suite does not exhaustively test every stated law; existential monotonicity, for example, is proved in Appendix B but not checked by that CLI.

Operator	Family	n	Surf.	Act.	Cls.
COMPLEMENT	equiv.	501	0	0	0
EXISTS	equiv.	501	0	0	0
HIERARCHY	equiv.	501	0	0	0
POLARITY	equiv.	501	0	0	0
PROJECTION	equiv.	501	0	0	0
ALIAS	inv.	505	0	0	0
PARAPHRASE	inv.	505	0	0	0
REORDER	inv.	505	0	0	0

Table 8: Algorithmic construct-validation ledger. "Surf./Act./Cls." count broken surface, action, and closure checks; all operators pass every check.

Human semantic audit. The audit described in §4.1 uses three internal research collaborators as annotators on $N{=}200$ realizations (25 per operator); each annotator sees the base prompt, the rewritten prompt, the task description, and the proposed action $g_{\tau,\omega}$ but no model output, and assigns one of three categories: "preserves the target," "applies the stated action," or "introduces additional requirements / misapplies the action." Overall pass rate is 92.5% $(185/200)$ and overall Fleiss' $\kappa{=}0.83$ across the eight operators. The two binary equivariance operators (POLARITY, COMPLEMENT) reach near-perfect agreement ($\kappa{\geq}0.90$), while the structured equivariance operators (EXISTS, HIERARCHY, PROJECTION) show agreement ($\kappa{=}0.74$–0.79), showing that agreement differs across operator families. The fifteen flagged realizations were discarded and replaced before any model was scored; the audit therefore affects the retained operator library rather than the model scores themselves.

Dataset	Parse fail	Score	Score | parse ok
GoEmotions	0.003	0.311	0.312
Few-NERD	0.081	0.176	0.203
MAVEN	0.019	0.022	0.022

Table 9: Structured-task scores before and after conditioning on successful canonicalization. Scores are mean $1 - R^{\star}$ over the 12 structured-task model–operator cells per dataset; higher is better. These are macro-averaged distance scores, not binomial proportions or base-correct conditional failure estimates.

Paired baseline tests on the full behavior run. The baseline suite behind Table 3 confirms that the main-text reading is not an accident of one summary table. On Qwen2.5-7B BoolQ, multi-prompt accuracy and InvScore are numerically close ($\Delta{=}{-}0.001$, paired bootstrap $p = 0.645$), a descriptive similarity that does not establish equivalence. On the same model and dataset, single-prompt accuracy still exceeds EqvScore by approximately 13 percentage points, showing that prompt-ensemble baselines do not expose the complement failure. On GoEmotions, the multi-prompt-average vs. InvScore gap remains positive ($\Delta{=}0.008$, $p < 10^{-4}$).

Structured-task failures are not parse-fail artifacts. For Qwen2.5-7B, parse-failure rates remain at or below 4.9% on Few-NERD, $\leq 1.5\%$ on GoEmotions, and $\leq 1.2\%$ on MAVEN, yet operator risks remain high (Figure 3).

C.2 Parse-conditioned structured-task analysis

Table 9 reports the mean structured-task score before and after conditioning on successful canonicalization, averaged over models and operators. The comparison supports the scoped reading in the main text: GoEmotions and MAVEN failures are not driven by parser failure, while Few-NERD is more sensitive to schema compliance, especially for Phi-4-mini's alias prompts.

Reading Figure 4. Figure 4 expands the behavior–diagnostic correspondence into a layerwise profile on Qwen2.5-7B and Qwen3-4B.

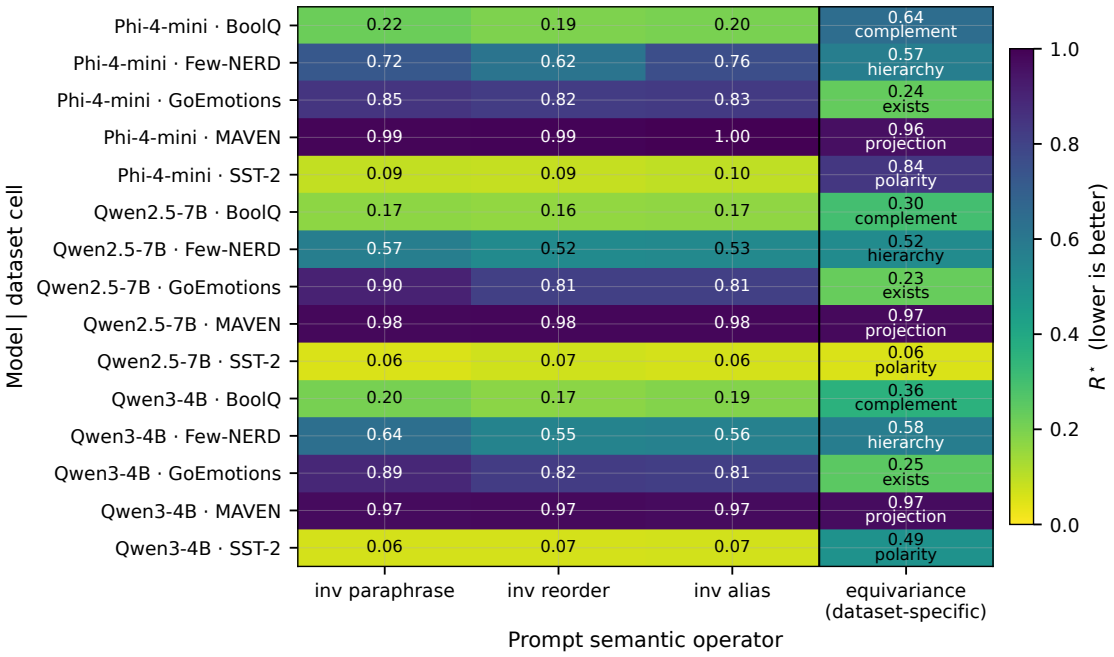


Figure 3: Per-operator $R^\star$ across models and datasets (lower is better). Rows are (model, dataset) cells; the left block aggregates the three invariance operators and the right column reports the dataset-specific equivariance operator. Equivariance risks depend on both the model and operator. The scoring space changes for structured equivariance operators, so cross-operator differences are descriptive and do not share a universal difficulty scale.

SLR and RSR quantify how much of the operator-induced activation shift δ_ω^ℓ lands in the task-readout subspace; RSR normalizes SLR by a random-direction projection baseline. RSR and NIS are descriptive diagnostics defined for this study, not causal estimands. NIS is $\mathbb{E}[\|\delta_\omega^\ell\|_2]/(\sigma^\ell+\varepsilon)$, where σ^ℓ is the mean within-class clean-state spread. ACI measures, on equivariance operators, whether the projected component points in the semantic direction predicted by $g_{\tau,\omega}$. ΔLID is the perturbed-minus-clean change in local intrinsic dimension. The supplied reference implementation uses a TwoNN-style estimator (Facco et al., 2017): Euclidean first/second-neighbor distance ratios, a default 10% upper-tail discard, and a regression through the origin on logarithmic ratios and empirical ranks. It re-ranks after trimming, a variant of the published empirical-CDF construction; we do not identify it with an unmodified TwoNN estimator. This implementation alone does not establish the historical settings behind each plotted curve. We use the difference as a descriptive geometric summary rather than a calibrated measure of semantic understanding.

Reading the Mapper visualization. Mapper (Singh et al., 2007) summarizes overlapping clusters in a filtered point cloud. Figure 5 visualizes the clean and perturbed hidden-state manifolds on SST-2 polarity. The clean and perturbed manifolds remain qualitatively similar in shape but shift along the probe-readout direction, consistent with a displacement in the selected representation. Together with Figure 4, these plots provide a qualitative geometric description of the sampled hidden states. They do not establish that fragmentation or displacement is the cause of a behavioral failure. We verified that this qualitative split persists across alternative lens functions (PCA vs. UMAP) and standard cover-overlap values.

C.3 OrbitAlign ablation and error decomposition

Table 10 compares four training conditions. *Base* is the unadapted Qwen2.5-7B. *Task-only* is a QLoRA adapter trained on just the 2,000 original task rows ($\lambda_{\text{inv}}{=}\lambda_{\text{eqv}}{=}0$), a control for adaptation without operator-augmented data. *Unpartitioned* is a QLoRA adapter trained on the full 10,000 operator-expanded rows (task + inv + eqv) with a single uniform token-level cross-entropy; every row contributes to the gradient equally but no family is weighted separately. *OrbitAlign* uses the same 10,000 rows but partitions them by family with $\lambda_{\text{task}}{=}1.0$, $\lambda_{\text{inv}}{=}\lambda_{\text{eqv}}{=}0.5$. The decisive compar-

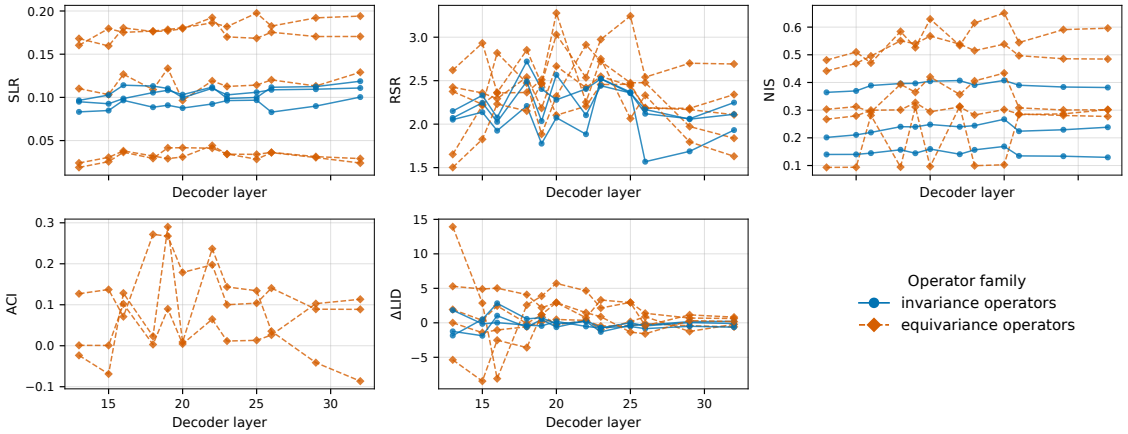


Figure 4: Layerwise representation diagnostics across Qwen2.5-7B and Qwen3-4B. SLR and ACI are defined in §D (Eqs. (8) and (9)); RSR, NIS, and ΔLID are defined in the preceding paragraph of this appendix.

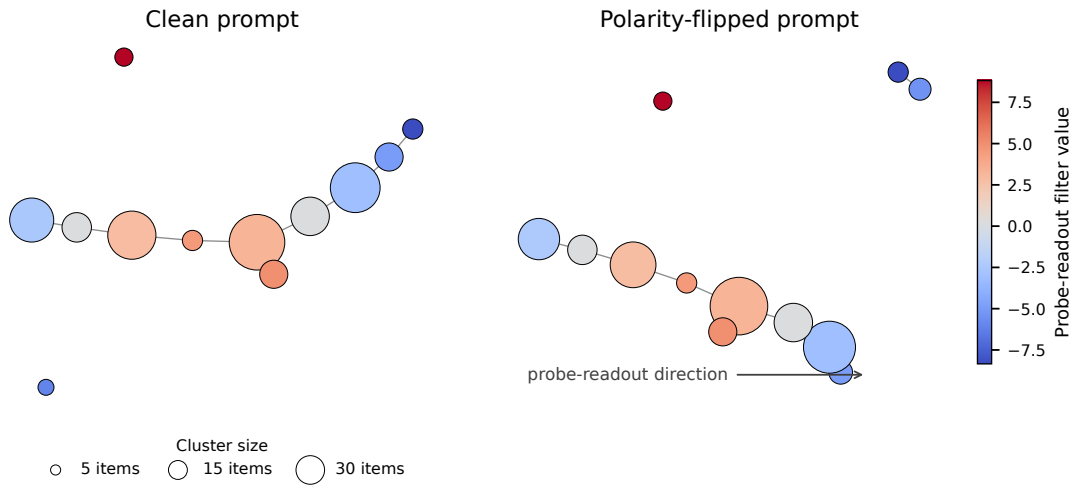


Figure 5: Mapper visualization of the SST-2 polarity hidden-state manifold. Clean (left) and perturbed (right) clouds shift along the probe-readout axis. Node color encodes the probe-readout filter value, node size encodes cluster size, and edges indicate cluster overlap.

ison is Unpartitioned vs. OrbitAlign: same data, same optimizer, same LoRA rank, only the loss decomposition differs. The results show a clean decomposition of effects: augmentation alone (Unpartitioned) closes most of the gap on binary tasks, while the family-partitioned objective (OrbitAlign) adds further gains on structured tasks—MAVEN projection drops a further 4.4 points, Few-NERD hierarchy a further 1.7 points. This is consistent with a diluted-signal interpretation: in richer canonical spaces the equivariance gradient is a small fraction of the total token loss under a uniform CE, and changing the relative contribution of the families can improve these structured-task scores. This interpretation is descriptive; the comparison does not directly measure gradient dilution. All four conditions are evaluated on the same matched ablation pool (the BoolQ row has $n = 200$). For BoolQ complement the paired reduction is $0.298 - 0.105 = 0.193$. This differs from the full repair pool ($n = 400$), where the paired reduction is $0.441 - 0.187 = 0.254$.

(Dataset, operator)	Base	Task	Uniform	Orbit
SST-2, polarity	0.055	0.470	0.040	0.040
BoolQ, complement	0.298	0.655	0.100	0.105
GoEmot., exists	0.233	0.205	0.195	0.190
Few-NERD, hierarchy	0.518	0.436	0.421	0.404
MAVEN, projection	0.974	0.915	0.761	0.717

Table 10: Equivariance-operator $R^\star$ (lower is better) under four conditions: *Task* is task-only QLoRA, *Uniform* is unpartitioned CE on all 10k rows, *Orbit* is OrbitAlign (partitioned); the reported BoolQ pool has $n = 200$. Unpartitioned already matches OrbitAlign on binary tasks; partitioning yields additional gains on structured tasks.

C.4 Held-out original-prompt accuracy

Table 11 reports the strict original-prompt exact-match check for the OrbitAlign adapter on a pool disjoint from the adapter training examples. The structured-task rows use exact set/field matching and are therefore not directly comparable to the behavior-layer aggregates in §5; the point is that the repair does not degrade clean-prompt behavior under the same strict metric.

Dataset	Base	OrbitAlign
SST-2	0.929	0.945
BoolQ	0.833	0.895
Few-NERD	0.000	0.240
GoEmotions	0.090	0.275
MAVEN	0.008	0.145

Table 11: Held-out original-prompt accuracy (zero-distance in the canonical space) for Qwen2.5-7B before and after OrbitAlign.

C.5 Held-out operator wordings

Table 12 evaluates the base model and OrbitAlign adapter on unseen natural-language realizations of the two binary equivariance actions. The examples are adapter-disjoint (400 training identifiers excluded per dataset), the semantic action remains $z \mapsto 1 - z$, and only the operator wording changes. This isolates template generalization from task memorization. The SST-2 operator is already easy for the base model under the new wording, and OrbitAlign preserves the low risk. BoolQ is the stricter test: the base model mostly answers the ordinary question instead of its complement, whereas OrbitAlign transfers the learned complement behavior to the held-out wording.

Dataset	Model	$R^\star$	R°	Orig. acc.
SST-2	Base	0.070	0.060	0.940
SST-2	OrbitAlign	0.045	0.040	0.945
BoolQ	Base	0.785	0.885	0.890
BoolQ	OrbitAlign	0.190	0.195	0.895

Table 12: Unseen-template equivariance evaluation on 200 adapter-disjoint examples per dataset. All parse-failure rates are 0.0%. Lower $R^\star$ and R° are better.

C.6 Structured-task identifiability

The conditional question is whether the rewritten output violates the declared action on an item whose clean prediction was correct. This requires enough correctly answered clean items; a large number of total evaluated examples is insufficient when clean accuracy is low. For the MAVEN projection diagnostic, the available clean-correct subset contains four items and no observed law failures: $0/4$. We report the exact count and treat the cell as underpowered. It does not establish that the projection action is reliably obeyed.

The conditional gate used for the binary control analysis requires parse success at least 98%, clean accuracy at least 75%, at least 200 clean-correct observations and a 95% confidence-interval half-width no larger than 7.5 percentage points. These requirements apply to the designated control analysis; they are not a claim that every earlier pool or every structured cell meets the gate. The structured datasets remain extensibility diagnostics, for which we report unconditional risks and parse-conditioned distance scores.

C.7 Frontier-class API audit on binary equivariance

Table 13 reports an adapter-disjoint black-box audit on three frontier-class instruction-tuned systems (DeepSeek-V3.2, Kimi-K2.5, Qwen3.5-397B-A17B) for the two binary equivariance operators. Each cell uses 200 adapter-disjoint examples. All API calls use temperature 0 with no system prompt; the user-turn prompt template is identical to the open-weights experiments and the raw text output is processed by the same deterministic canonicalizers and distance functions. All cells exhibit zero parse failures. Because API providers may change model weights or serving infrastructure without notice, the results should be read as a cross-model snapshot rather than a guaranteed reproducible ordering or an isolated effect of model scale.

On BoolQ complement, the three audited systems have gold-grounded risks 0.175, 0.140 and 0.070 despite clean accuracies of 0.955, 0.970 and 0.955. SST-2 polarity risks range from 0.035 to 0.115. Architecture, training and serving differ across systems, so the comparison does not identify an effect of parameter count. The open-weights binary controls provide the controlled evidence; these six API cells extend the descriptive scope.

Model	Op.	$R^\star$	R°	Acc.
DeepSeek-V3.2	pol.	0.115	0.130	0.940
Kimi-K2.5	pol.	0.055	0.035	0.950
Qwen3.5-397B	pol.	0.035	0.010	0.955
DeepSeek-V3.2	cmp.	0.175	0.150	0.955
Kimi-K2.5	cmp.	0.140	0.150	0.970
Qwen3.5-397B	cmp.	0.070	0.045	0.955

Table 13: Frontier-class API audit on 200 adapter-disjoint examples per cell; "pol."/"cmp." denote polarity/complement and "Acc." is original-prompt accuracy. All parse-failure rates are 0.0%. Values are a cross-model snapshot; parameter scale is not isolated by this comparison.

C.8 Cross-backbone OrbitAlign replication

To test whether the OrbitAlign objective is specific to the Qwen2.5-7B backbone reported in the main paper, we apply the *identical* recipe (λ_{task}=1.0, λ_{inv}=λ_{eqv}=0.5, LoRA r=16, α=32, 400 base items per dataset, AdamW lr=2×10^{-4}, batch 2, gradient accumulation 8, one epoch, 4-bit NF4) to two further backbones spanning the Qwen and Phi families: **Qwen3-4B-Instruct-2507** and **Phi-4-mini-instruct**. The example pool, train ID manifest, canonicalizers, distance functions, and held-out evaluation protocol follow the same evaluation recipe; the LoRA target modules change where required by the architecture. Each reported cell uses 200 example-ID-disjoint items. Table 14 reports per-cell behavior on both backbones. Across the 20 Qwen3-4B cells every cell shows $\Delta R^\star \le 0$ (largest reduction 0.440); across the 20 Phi-4-mini cells, 19/20 cells show $\Delta R^\star \le 0$ (largest reduction 0.780 on SST-2 polarity). The only positive shift is +0.015 on GoEmotions EXISTS; this point estimate is a small regression, and its statistical uncertainty is not resolved by the table. Strict original-prompt clean accuracy rises on every dataset for both backbones (Qwen3-4B: SST-2 0.950→0.960, BoolQ 0.875→0.930, GoEmotions 0.115→0.250, Few-NERD 0.180→0.275, MAVEN 0.005→0.170; Phi-4-mini: SST-2 0.915→0.945, BoolQ 0.855→0.865, GoEmotions 0.085→0.215, Few-NERD 0.085→0.285, MAVEN 0.005→0.145). The repair pattern therefore replicates across two distinct backbones from two model families, supporting the claim that operator-law violations are amenable to lightweight QLoRA repair across backbones rather than being specific to a single model.

Cross-family interpretation. The largest binary-flip reductions in the displayed replication are 0.440 and 0.310 on Qwen3-4B, and 0.780 and 0.580 on Phi-4-mini. The comparison supports replication of the training recipe across two model families. It does not establish transfer of the same learned adapter between architectures, unrestricted transfer to unseen actions, or significance for every cell.

Held-out clean accuracy. Beyond the operator-law repair, OrbitAlign preserves or improves held-out clean-prompt accuracy on every dataset for both backbones, providing a separate check on clean-prompt behavior. Table 15 collects the before/after clean accuracies on the held-out evaluation pool: every cell satisfies $\text{Acc}_a \ge \text{Acc}_b$, with gains concentrated on the structured tasks where the base accuracy was weakest.

Scope of the repair evidence. The later aggregate evaluation records non-increasing gold risk in 59/60 cells across the three backbones. The two additional-backbone tables here provide the explicit 39/40 component. This is a sign count, not a claim of 59 significant improvements. The recipe was fixed on Qwen2.5-7B and replicated without per-backbone hyperparameter tuning. The method

Data	Op.	$R_b^\star$	$R_a^\star$	$\Delta R^\star$	ΔR°
(a) Qwen3-4B (Qwen family)					
sst2	paraphr.	0.050	0.035	−0.015	+0.000
	reorder	0.055	0.040	−0.015	−0.030
	alias	0.045	0.035	−0.010	−0.015
	polarity	0.495	0.055	−0.440	−0.455
boolq	paraphr.	0.160	0.070	−0.090	−0.025
	reorder	0.110	0.070	−0.040	−0.025
	alias	0.135	0.075	−0.060	−0.005
	complement	0.400	0.090	−0.310	−0.485
goem.	paraphr.	0.878	0.727	−0.151	−0.510
	reorder	0.773	0.715	−0.057	−0.139
	alias	0.765	0.724	−0.041	−0.117
	exists	0.235	0.195	−0.040	−0.145
f.nerd	paraphr.	0.656	0.452	−0.204	−0.256
	reorder	0.515	0.439	−0.076	−0.113
	alias	0.539	0.451	−0.089	−0.156
	hierarchy	0.535	0.400	−0.136	−0.027
maven	paraphr.	0.986	0.613	−0.373	−0.506
	reorder	0.988	0.622	−0.366	−0.546
	alias	0.986	0.626	−0.360	−0.284
	projection	0.979	0.690	−0.289	−0.122
(b) Phi-4-mini-instruct (Phi family)					
sst2	paraphr.	0.085	0.050	−0.035	−0.005
	reorder	0.065	0.050	−0.015	−0.035
	alias	0.085	0.060	−0.025	−0.005
	polarity	0.825	0.045	−0.780	−0.810
boolq	paraphr.	0.190	0.140	−0.050	−0.020
	reorder	0.145	0.125	−0.020	−0.020
	alias	0.160	0.145	−0.015	−0.005
	complement	0.715	0.135	−0.580	−0.760
goem.	paraphr.	0.812	0.710	−0.102	−0.269
	reorder	0.775	0.719	−0.056	−0.172
	alias	0.793	0.723	−0.070	−0.181
	exists	0.200	0.215	+0.015	−0.090
f.nerd	paraphr.	0.763	0.464	−0.299	−0.505
	reorder	0.672	0.445	−0.227	−0.266
	alias	0.790	0.467	−0.323	−0.491
	hierarchy	0.597	0.427	−0.170	−0.286
maven	paraphr.	0.997	0.717	−0.280	−0.565
	reorder	0.997	0.724	−0.273	−0.612
	alias	1.000	0.715	−0.285	−0.677
	projection	0.983	0.741	−0.242	−0.253

Table 14: Cross-backbone OrbitAlign replication, $n = 200$ items per dataset and backbone. Subscripts b/a denote base/adapter; $\Delta R = R_a - R_b$, so negative values are improvements. Qwen3-4B improves on $20/20$ cells and Phi-4-mini on $19/20$. Deltas were rounded separately from risks; a displayed subtraction can therefore differ by 0.001. Point-estimate reductions do not imply a statistically significant change in every cell.

Dataset	Qwen3-4B		Phi-4-mini	
	before	after	before	after
SST-2	0.950	0.960	0.915	0.945
BoolQ	0.875	0.930	0.855	0.865
GoEmotions	0.115	0.250	0.085	0.215
Few-NERD	0.180	0.275	0.085	0.285
MAVEN	0.005	0.170	0.005	0.145

Table 15: Cross-backbone held-out original-prompt clean accuracy before/after OrbitAlign on the adapter-disjoint evaluation pool, $n = 200$ items per dataset and backbone. Every cell satisfies $\mathrm{Acc}_a \geq \mathrm{Acc}_b$, describing clean-accuracy preservation in these full-data replication conditions. The separate transfer runs have small clean-accuracy decreases, reported in Table 18.

remains supervised learning on operator-expanded targets; its generalization evidence should be read at the stated task, wording and action-family level.

C.9 Non-deceptive controls and mitigation baselines

The control experiment evaluates SST-2 polarity and BoolQ complement on Phi-4-mini and Qwen2.5-7B using 400 items per model–task cell and five wordings. Neutral codebooks identify outputs as codes (A/B or X/Y), not factual assertions. A dual-field prompt asks for both an `ordinary_answer` and an `operator_output`. An operator-only oracle supplies the clean label and asks only for the mapping. Reported intervals use item×wording bootstrap resampling; values below summarize the control cells.

Control summary	Value
Dual-field ordinary-accuracy change	-1.1 pp
Range across control cells	$[-1.8, -0.6]$ pp
Codebook gap vs. clean accuracy	17.8 pp
Law failure given ordinary correctness	14.6%
Explicit refusal among violations	2.8%
Ordinary-answer echo among violations	67%
Operator-only oracle compliance	97.4%
Joint dual-field compliance	72.1%
Oracle minus joint compliance	25.3 pp

Table 16: Non-deceptive control summaries; larger compliance is better. The 17.8-point gap has a 95% CI of $[12.4, 23.1]$ pp; the conditional 14.6% rate has a CI of $[9.8, 19.7]\%$. Refusal and echo percentages use violation events as their denominator; the conditional law-failure rate uses ordinary-correct predictions. The total sample budget is 400 per cell; exact conditional counts are not specified in the aggregate report.

The ordinary answer remains largely intact while a residual mapping gap survives non-deceptive framing. Thus honesty alone does not explain the measured failures. This does not make refusal intrinsically undesirable: the scored object is adherence to the declared output contract, and refusal is recorded separately from answer echo.

One-call decomposition improves compliance by 22.6 percentage points but retains an 18.1-point gap against clean accuracy (95% CI $[13.0, 23.4]$ pp). A two-call interface or external deterministic

Condition	Compliance
Direct one-shot	38.6%
Explicit one-call two-step	61.2%
Two-call infer → transform	96.8%
Deterministic mapping of a supplied label	100%
OrbitAlign, held-out wording, one call	91.4%

Table 17: Mitigation ladder on the binary control cells (400 items per cell, five wordings). Higher compliance is better. The deterministic rung measures exact execution of the specified mapping on its input label; it is not 100% transformed-gold task accuracy.

mapping can therefore be a practical mitigation where the interface permits it. OrbitAlign addresses the separate constraint of retaining a single model call.

C.10 Same-action transfer and composition checks

Transfer keeps the action $z \mapsto 1-z$ while changing the task and holding out example IDs and wordings. The following paired changes are adapted minus base; negative risk changes are improvements.

Matched task-only and token-matched augmentation controls change risk by approximately -0.01 and -0.02, respectively. The full-data BoolQ repair gives the larger 0.254 reduction. Additional leave-one-out and composition checks improve $11/14$ evaluated cells, and a three-seed check reports agreement in direction with standard deviation below 0.03. These aggregate checks do not identify an unrestricted unseen-operator-family capability; the reported transfer is strongest for the shared binary-flip action.

A separate pure-action versus joint-task diagnostic reports 96.1% and 74.8%, a 21.3-point composition gap. This summary is not the same condition as the $97.4\%/72.1\%$ oracle/dual-field comparison above. Its aggregate record does not identify the individual task–model cells or denominators, so it supplies descriptive corroboration rather than a separate inferential claim.

D Representation Diagnostics

Observed self-grounded violations motivate representation diagnostics: the next question is whether the perturbation leaks into the task-relevant representation subspace and whether an equivariant perturbation moves in the direction predicted by its semantic action. We probe Qwen2.5-7B and Qwen3-4B at mid-to-late decoder layers on held-out subsamples. The reference configuration specifies inclusive, 1-indexed layer ranges 12–28 for Qwen2.5 and 10–24 for Qwen3, a linear probe with L2 coefficient 10^{-3} and 10,000 maximum iterations, and Mapper cover parameters of eight cubes and 0.4 overlap. Diagnostic sample IDs, per-curve execution records, and the selected patch layers are absent, so these configuration values do not certify every plotted run. The reference SLR and ACI functions use $\varepsilon = 10^{-8}$.

Let $h^\ell(P,T)$ denote the hidden state of the decoder at mid-to-late layer ℓ on the clean prompt P applied to text T, and let W^ℓ denote a linear probe trained on these clean-prompt hidden states to predict $z^\star_\tau(T)$. Let Π_{W^ℓ} be the orthogonal projection onto the row space of W^ℓ (the *task-readout subspace*), and let $\delta^\ell_\omega = h^\ell(\omega(P),T) - h^\ell(P,T)$ be the activation shift induced by the operator ω. We measure the *Semantic Leakage Ratio*

$$\mathrm{SLR}^\ell_\omega = \mathbb{E}\left[\frac{\|\Pi_{W^\ell}\delta^\ell_\omega\|_2}{\|\delta^\ell_\omega\|_2 + \varepsilon}\right], \tag{8}$$

which describes the fraction of the shift in the probe row space; small values are compatible with readout stability but are not required by behavioral invariance, and the *Action Consistency Index*

$$\mathrm{ACI}^\ell_\omega = \mathbb{E}\Big[\cos\big(\Pi_{W^\ell}\delta^\ell_\omega, \Delta^\ell_\omega(z^\star)\big)\Big], \tag{9}$$

where $\Delta^\ell_\omega(z^\star) = \mu^\ell_{g_{\tau,\omega}(z^\star)} - \mu^\ell_{z^\star}$ is the centroid-based target direction: the difference between the clean-prompt hidden-state centroid μ^ℓ_z at class z and the corresponding centroid at the operator image $g_{\tau,\omega}(z)$. This centroid construction is directly defined for binary actions whose source and target classes share a clean-state representation space. For a type-changing action, a target centroid requires an additional representation definition; we do not infer one from a label-space projection. ACI close to one indicates alignment with the chosen direction, not proof of a causal mechanism. The supplied ACI helper accepts one target direction per batch; the stated expectation requires grouping by target direction or an item-specific wrapper, which is not included in the archive.

The structured tasks have larger SLR and larger gold-grounded invariance risks than the binary tasks in Table 19. This is a descriptive cross-task association: task difficulty and readout dimensionality can confound it. For binary equivariance, SST-2 polarity ACI is 0.153 on Qwen2.5-7B and 0.142 on Qwen3-4B, indicating a weak positive alignment

Training → evaluation	Base $R^\star$	Adapted $R^\star$	$\Delta R^\star$ (95% CI)	Clean Acc. change
SST-2 polarity → BoolQ complement	.482	.331	$-.151$ $[-.194, -.107]$	-0.3 pp
BoolQ complement → SST-2 polarity	.518	.389	$-.129$ $[-.171, -.088]$	-0.5 pp

Table 18: Same-action cross-task transfer. Each difference compares base and adapted predictions within its transfer pool. The intervals are the reported paired intervals; the aggregate record does not specify the transfer sample counts. These are separate conditions from the full-data repair and matched four-condition ablation.

Dataset	$\overline{\text{SLR}}$ (inv)	$\overline{R^\star}$ (inv)
SST-2	0.035	0.065
BoolQ	0.022	0.165
Few-NERD	0.106	0.541
GoEmotions	0.159	0.842
MAVEN	0.165	0.980

Table 19: Mean Semantic Leakage Ratio and mean behavioral invariance risk $R^\star$, averaged over the three invariance operators on Qwen2.5-7B. SLR is averaged over selected diagnostic layers and the three invariance operators; behavioral risk is averaged over those operators. Larger values on structured tasks describe a cross-task association, not a causal effect of the representation shift.

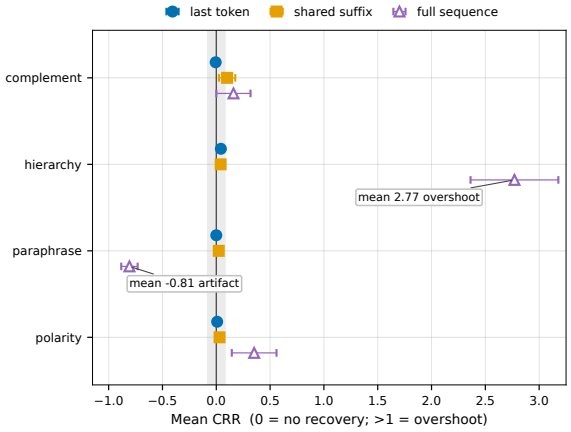


Figure 6: Operator-level mean Causal Recovery Ratio for the three patching strategies. Principled alignments (LAST-TOKEN, SHARED-SUFFIX) stay near zero, indicating little recoverable single-layer effect under aligned interventions. FULL-SEQUENCE can inflate or deflate recovery when prompt lengths differ, so it functions as a negative control rather than an informative intervention.

with the chosen centroid direction. The extended geometry and Mapper plots in Appendix C are supporting descriptions. They neither localize a circuit nor establish a causal explanation of the behavioral error.

E Patching Controls

E.1 Alignment-sensitive patching

We evaluate three patching strategies: LAST-TOKEN (decision state only), SHARED-SUFFIX (patch only positions with exact token agreement), and FULL-SEQUENCE (truncate-and-pad every position). The last is treated as a negative control whenever clean and perturbed prompts differ in length. For each strategy we patch the top-3 layers suggested by the mechanism sweep and report the Causal Recovery Ratio

$$\text{CRR} = \frac{R^\star_{\text{pert}} - R^\star_{\text{patched}}}{R^\star_{\text{pert}} - R^\star_{\text{orig}} + \varepsilon},$$

where $\varepsilon = 10^{-3}$ stabilizes the ratio when the gap is small. Intuitively, $\text{CRR} = 1$ means the patch approximately restores the clean-prompt risk when the stabilizer is negligible, $\text{CRR} = 0$ means the patch has no recovery effect, when the perturbed-minus-clean gap is positive. Negative or above-one values are sensitive to the sign and size of that gap; for principled alignments we treat values close to zero as the honest outcome.

Figure 6 supports a methodological conclusion: *prompt-length-mismatched patching requires an alignment-aware control*. Across the 11 core patching targets, principled alignments stay near zero CRR at the decision state, while FULL-SEQUENCE inflates or deflates recovery to artifactual values (mean CRR 2.77 on hierarchy and -0.81 on paraphrase). These results motivate alignment controls and caution with normalized recovery ratios. The near-zero aligned results concern these interventions and selected layers; they do not rule out causal features elsewhere. OrbitAlign uses distributed parameter-efficient adaptation rather than a claim of single-circuit localization.