The Illusion of Control: Why Bare Classifier Inversion Silently Fails in Concept-Bottleneck Text Generation

Qi Bing
Shanghai Jiao Tong University
peach19981011@gmail.com

Xiaowei Shao
Shanghai Jiao Tong University
xw.shao@sjtu.edu.cn

Abstract

Concept-bottleneck controllable generation routes multi-attribute control through a low-dimensional concept code that, at deployment, must be synthesised from a target attribute configuration. We study this problem in concept-bottleneck text generation under multi-axis compositional generalisation, comparing three ways to obtain the inference-time code: classifier inversion against the encoder heads, reference-text encoding, and a post-hoc label-conditioned prior. Since a concept code admits no direct LM-fluency term, regularising inversion must instead constrain the code toward the encoder's training distribution. We therefore test bare inversion and three regularised variants: label-agnostic and label-conditioned Mahalanobis penalties, and a conditional normalising-flow density baseline. Every inversion variant we test underperforms a simple post-hoc prior fitted to per-combination encoder means on the same checkpoints, across three backbone families spanning 124M to 8B parameters. The bare form of classifier inversion also silently collapses to chance, traceable to a directly measured off-manifold code. We validate this diagnosis on real-world benchmarks and under external evaluators, enabling fair comparison with published baselines.

1 Introduction

Multi-attribute controllable text generation (MCTG) requires generating text that satisfies several attribute axes $\mathbf{c} = (c_1, \ldots, c_A)$, and is commonly evaluated through compositional generalisation, where performance is measured by accuracy on attribute combinations held out during training (Keysers et al., 2020; Zhong et al., 2024). Recent concept-bottleneck language models (CB-LLMs) (Sun et al., 2025) show that language generation can be routed through interpretable concept units and steered by intervening on them, making the CB-LLM design a natural starting point for concept-bottleneck MCTG. In such a setting, deployment requires synthesising a code $\mathbf{z}^\star$ from a target configuration $\mathbf{c}^\star$. Since the bottleneck exposes classifier heads over $\mathbf{z}$, a natural inference path is to optimise $\mathbf{z}$ against those heads. The broader CTG steering literature typically combines such classifier gradients with fluency, prototype-distance, or manifold regularisation (Dathathri et al., 2020; Gu et al., 2022, 2023). Unlike token-level or decoding-time steering, however, optimisation over a concept code has no direct LM-fluency term; any regularisation must act through the code distribution itself. As a result, the role and failure mode of classifier-based code optimisation are often hidden inside compound inference objectives. We therefore isolate this component in the concept-bottleneck setting and ask what happens when the target attributes are read out directly through the bottleneck classifier heads.

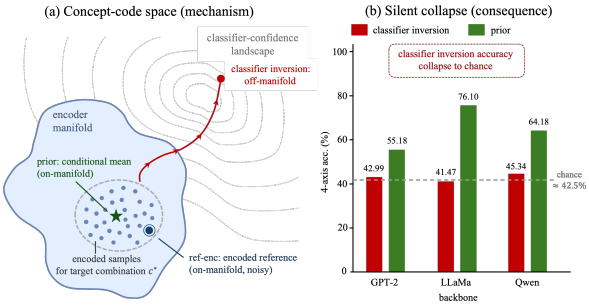

Figure 1: **The illusion of control. (a)** In concept-code space, classifier inversion drifts off the encoder's training manifold; a label-conditioned prior sits at the on-manifold per-combination mean, and `ref-enc` is one on-manifold sample. **(b)** CompMCTG Fyelp 4-axis accuracy: `cls-inv` sits at the $\approx 42.5\%$ chance baseline while the prior recovers compositional control on the *same* checkpoints (split details: Table 1 †).

We isolate this component experimentally (Fig. 1) and identify a directly measured failure mechanism: the inverted code lies 3 to 7 times far-

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 45311–45333
October 24-29, 2026 ©2026 Association for Computational Linguistics

ther from the encoder's training distribution than a working code, both in diagonal-Gaussian Mahalanobis distance and in mean distance to its 10 nearest training neighbours. Adding a manifold regulariser to the inversion objective lifts accuracy off chance (§5). The corrective protocol is a single forward pass: freeze the trained encoder and fit a post-hoc MLP g_γ to the per-combination means of the encoded training data. Given the target labels, the prior model estimates the conditional mean of the encoder code. Because the within-combination variation we measure exceeds the across-combination signal, the prior acts as a denoiser of sample-specific code variation, explaining its advantage over single-sample reference encoding (§6.2).

This paper makes three contributions.

- We systematically compare classifier inversion against a simple post-hoc label-conditioned prior on matched checkpoints. We test bare inversion and three regularised inversion variants (label-agnostic Mahalanobis, label-conditioned Mahalanobis, and a conditional normalising-flow density baseline) and find that every tested inversion variant underperforms the prior by 7 to 29 pp (Tab. 2; Apps. I, K). The bare form also collapses to chance, which we trace to a directly measured off-manifold code (§5).

- We introduce a deployable post-hoc label-conditioned prior that recovers compositional generalisation on the same checkpoints, with no inference-time optimisation and no reference text. The prior is fitted on our proposed multi-axis concept-bottleneck architecture that extends existing single-axis steering designs. To our knowledge, this is the first concept-bottleneck text-generation protocol evaluated under multi-axis compositional generalisation (§6, §7).

- We show that the prior acts as a conditional-mean denoiser, and provide evidence that the non-additive residual of the concept code is dominated by per-sample encoder noise rather than measurable attribute interaction (§6.2).

2 Related Work

Concept-bottleneck language models. Concept-bottleneck models constrain output through a low-dimensional, interpretable code, originating in vision (Koh et al., 2020) and recently adapted to language as text-classification bottlenecks (Tan et al., 2024; Sun et al., 2024; Bhan et al., 2025; Labadie-Tamayo et al., 2025; Chaudhary et al., 2025) and, for generation, as CB-LLMs (Sun et al., 2025), which evaluate single-axis steering on each dataset. To our knowledge, this paper is the first to extend the per-axis bottleneck from single-axis steering to multi-axis compositional generalisation on combinations unseen during training — the setting in which the inference-protocol pathology surfaces (§5).

Compositional and multi-attribute CTG. Multi-attribute CTG includes joint-training methods (Keskar et al., 2019; Yang et al., 2023; Qian et al., 2022; Zeng et al., 2023), decoding-time classifier-gradient or logit-biasing methods (Dathathri et al., 2020; Yang and Klein, 2021; Krause et al., 2021; Liu et al., 2021), latent-space control (Gu et al., 2022, 2023), adapter fusion (Roy and Mishra, 2024), hidden-state edits (Kumar et al., 2023), token-level RL (Li et al., 2024), and learning-free FFN reweighting (Feng et al., 2024). Closest to our prior is Gu et al. (2023), who map an encoder posterior to a Gaussian via a normalising flow. Instead, we fit a deterministic post-hoc label-to-latent MLP without an invertibility constraint. Our generator is conditioned through hidden-state injection, a mechanism related to FiLM (Perez et al., 2018), AdaLN (Peebles and Xie, 2023), prefix-tuning (Li and Liang, 2021), and adapters (Houlsby et al., 2019), and implemented with AdaLN-zero and rank-8 LoRA (Hu et al., 2022). We evaluate on CompMCTG Fyelp (Zhong et al., 2024), built from the Maximum Compound Divergence split (Keysers et al., 2020), under its official RoBERTa-large evaluator.

Inference protocols for concept-conditioned generation. Inference-time construction of the concept code remains under-specified in concept-bottleneck generation. Existing classifier-gradient methods such as PPLM (Dathathri et al., 2020), classifier-guided diffusion (Li et al., 2022; Liu et al., 2023), and activation steering (Oozeer et al., 2025; Li et al., 2023) optimise control representations with additional fluency, prototype-distance, or distributional regularisation (Dathathri et al., 2020; Gu et al., 2022, 2023). In contrast, our focus is the inference protocol exposed by a trained concept bottleneck. We compare several protocol choices on matched checkpoints and find that the post-hoc

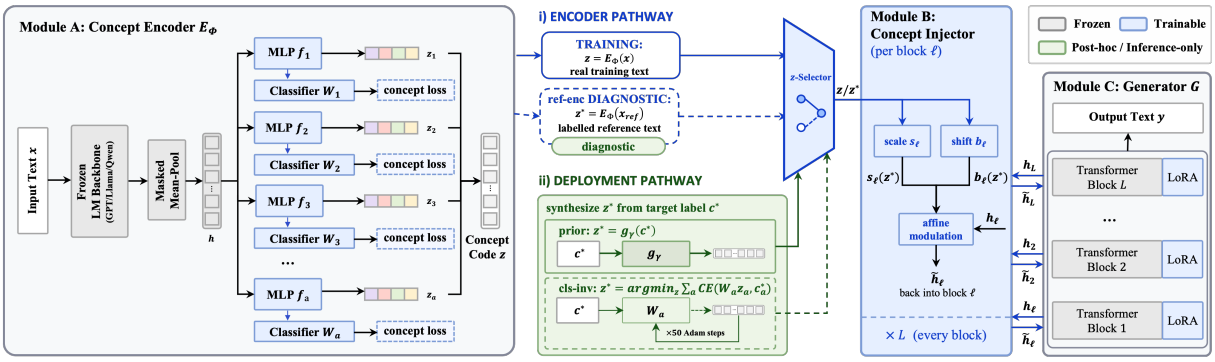

Figure 2: Architecture of the concept-bottleneck CTG framework (§3.1). **Module A** encodes input text $\mathbf{x}$ into a concatenated concept code $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_A)$ using a frozen-LM encoder, per-axis MLPs, and classifier heads. **z-Selector** supplies the injector input: during training, $\mathbf{z} = E_\phi(\mathbf{x})$; during inference, $\mathbf{z}^\star$ is obtained by one of the protocols in §3.2 (`prior`, `cls-inv`, or `ref-enc`). **Module B** maps the selected code to per-layer AdaLN-zero updates. **Module C** is a LoRA-adapted frozen Transformer generator that emits $\mathbf{y}$ autoregressively.

prior consistently dominates the tested inversion variants. Other steering work constructs control vectors without classifier inversion, e.g., by mean-difference or contrastive construction (Hsu et al., 2026; Lee et al., 2025; Stolfo et al., 2025).

3 Background

Since our target is an inference protocol, we first fix the object of study: a multi-axis concept-bottleneck architecture for compositional CTG, extending the single-axis concept-bottleneck generation design of Sun et al. (2025). This architecture poses the inference problem we study: synthesising an inference-time concept code from a target attribute configuration. It serves as the experimental vehicle for our protocol diagnosis.

3.1 Framework

Problem formulation. A controllable text generation task specifies A attribute axes, each with a finite value set $\mathcal{A}_a$. A *concept configuration* is $\mathbf{c} = (c_1, \ldots, c_A) \in \mathcal{A} := \prod_a \mathcal{A}_a$. Given $\mathbf{c}$, the model generates text $\mathbf{y}$ that satisfies all specified attributes, modelling $p_\theta(\mathbf{y} \mid \mathbf{c})$. During training, the generator is teacher-forced on $\mathbf{x}$ in an autoencoding setup, with $\mathbf{y} := \mathbf{x}$. Following the MCD protocol (Keysers et al., 2020; Zhong et al., 2024), training covers only $\mathcal{C}_{\text{seen}} \subset \mathcal{A}$; we report performance on both seen and unseen combinations, where $\mathcal{C}_{\text{unseen}} := \mathcal{A} \setminus \mathcal{C}_{\text{seen}}$.

Architecture. The architecture has three modules connected through continuous concept vectors (Fig. 2). First, a *concept encoder* $E_\phi : \mathcal{X} \to \mathbb{R}^{Ad_c}$ mean-pools a frozen backbone representation and passes it through A per-axis MLPs f_a, producing sub-codes $\mathbf{z}_a \in \mathbb{R}^{d_c}$ with per-axis classifier heads $\mathbf{W}_a$. The full concept code is $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_A)$. Second, a *concept injector* I_ψ maps the code to per-layer gated residual updates for the generator blocks; our default implementation uses AdaLN-zero (Peebles and Xie, 2023). Third, the *generator* G is a frozen pretrained Transformer adapted with LoRA (Hu et al., 2022). During training, the injector receives the encoder code $\mathbf{z} = E_\phi(\mathbf{x})$ and maps it to per-layer updates $\{\Delta\mathbf{h}_\ell\}$. At inference, the encoder path is bypassed, so an inference-time code $\mathbf{z}^\star$ must be synthesised from the target configuration $\mathbf{c}^\star$ (§3.2). Training uses a teacher-forced LM loss, a per-axis concept-classification loss, and an inter-axis orthogonality regulariser; the injector update and four-phase training schedule are given in App. A.

3.2 Inference Protocols

Any procedure that maps a target configuration $\mathbf{c}^\star$ to an inference-time code $\mathbf{z}^\star$ defines a distinct evaluation protocol. We compare three code sources.

Classifier inversion (CLS-INV). Given the exposed classifier heads, the direct classifier-based protocol optimises the code so that the heads predict the target labels:

$$\mathbf{z}^\star_{\text{cls}} = \arg\min_{\mathbf{z}} \sum_a \text{CE}(\mathbf{W}_a \mathbf{z}_a, c^\star_a). \quad (1)$$

The objective decouples over axes. We solve each axis with 50 Adam steps from $\mathbf{z}_a^{(0)} = \mathbf{W}_a[c^\star_a, :]^\top$.

Reference-text encoding (REF-ENC). As a single-sample diagnostic, we encode one labelled held-out example: $\mathbf{z}^\star_{\text{ref-enc}} := E_\phi(\mathbf{x}_{\text{ref}})$, where

$\pi_a(\mathbf{x}_{\text{ref}}) = c_a^\star$. We do not treat this as an upper bound. The encoded reference carries the surface idiosyncrasies of one sentence, so a conditional-mean estimator can outperform it by averaging away sample-specific variation. We include REF-ENC alongside PRIOR to make this variance source explicit (§6.2); it is not deployable, since it requires a labelled reference sentence matching the requested configuration.

Amortised label prior (PRIOR). After the main model has been trained, we freeze E_ϕ and fit a post-hoc MLP $g_\gamma : \mathcal{A} \to \mathbb{R}^{Ad_c}$ by

$$\mathcal{L}(\gamma) = \mathbb{E}_{(\mathbf{x},\mathbf{c})\sim\mathcal{D}_{\text{train}}} \left\| g_\gamma(\mathbf{c}) - E_\phi(\mathbf{x}) \right\|_2^2. \quad (2)$$

At inference, we use $\mathbf{z}^\star_{\text{prior}} := g_\gamma(\mathbf{c}^\star)$. The MLP g_γ has a single hidden layer of 128 units (GELU) and is fitted post-hoc in well under a minute. It is trained only on $\mathcal{C}_{\text{seen}}$ but queried on $\mathcal{C}_{\text{unseen}}$. The encoder's per-axis structure lets the learned label-to-code map provide a compositional estimate of the per-configuration encoder mean.

4 Experimental Setup

Our evaluation centres on CompMCTG Fyelp, a real multi-attribute review benchmark with compositional splits that supports the main claims. We also use a synthetic 4-axis task and single-axis YelpP as controlled checks.

Datasets. Our primary setting is **Fyelp** (CompMCTG, 4-axis sentiment × gender × cuisine × tense; 65K/1.5K/1.5K/1,750 train/val/`test_seen`/`test_unseen`) under two compositional splits: Hold-Out idx=−0 (39 seen and 1 unseen combination) and **ACD** (half of the 40 combinations unseen; §7.1). We also evaluate on **Amazon** (CompMCTG, 2-axis sentiment × topic; App. C), and use two controlled checks: a **Synthetic 4-axis MCD** task (4^4 combinations, 200 seen / 56 unseen via the MCD protocol (Keysers et al., 2020); §5–§6) and **YelpP** (binary sentiment, single-axis CB-LLMs cross-check; App. B).

Backbones. Tab. 1 reports four Fyelp backbones spanning 124M–1.5B: GPT-2 124M, GPT-2-Medium 355M (the backbone every CompMCTG baseline uses, and therefore our matched comparison in §7.1), LLaMA-3.2 1B (primary scale), and Qwen-2.5 1.5B. The first three use AdaLN-zero; Qwen-2.5 1.5B uses the additive injector after AdaLN-zero proved unstable. The synthetic sanity check (§5) adds Qwen-2.5 0.5B and LLaMA-3.2 3B, and a LLaMA-3 8B YelpP checkpoint covers the single-axis check. All models are LoRA-adapted on frozen weights (rank 8, $\alpha = 16$), trained for 25 epochs with AdamW and a cosine schedule, with concept dimension $d_c = 32$.

Evaluation. For Fyelp, we use the official CompMCTG 4-axis RoBERTa-large classifier suite (Zhong et al., 2024), invoked verbatim through a subprocess wrapper to match the published baseline pipeline; perplexity is computed with GPT2-large under the same pipeline. We never use the bottleneck encoder's own classifier heads for evaluation. We report per-axis and 4-axis-mean accuracy on generated text, joint all-axes-correct accuracy, and fluency metrics (perplexity and Dist-n); headline numbers are on `test_unseen`, except for CLS-INV on Hold-Out, which we report on `test_seen` because the single held-out combination admits a classifier-default artefact (App. Q); the ACD split gives the artefact-free within-split comparison. Tab. 1 marks the affected cells. For comparability, we fix one validation-selected stable recipe per backbone and a fixed evaluation seed 42; the full seven-point comparability policy is in App. O.

Artefacts. Every benchmark, evaluator, and backbone we use is a public third-party release under its own licence. We release our code under the MIT Licence, together with the trained concept encoder, classifier heads, injector, LoRA adapter, and label prior for all four main-table backbones on both splits.[1]

5 Classifier inversion silently collapses across backbones

Classifier inversion is the most direct classifier-based answer to the inference problem in §3.2. We isolate this procedure from the regularised objectives that usually surround classifier-gradient methods (§2) and find that it silently fails: on every trained checkpoint, it produces text with little or no concept signal while training metrics remain healthy. Through direct measurement and regulariser ablations, we trace the collapse to a consistent mechanism: classifier inversion drives the control code off the encoder's training distribution.

[1] https://github.com/BiancaBing/cbctg-illusion-of-control

5.1 The collapse: chance-level accuracy on every backbone

We applied classifier inversion (Eq. 1) to every trained checkpoint and evaluated the generated text under the official CompMCTG Fyelp evaluator. On the seen split (39 combinations), which we headline for CLS-INV for the reason given in §4, 4-axis accuracy is 41.47% on LLaMA-3.2 1B, 42.99% on GPT-2 124M, 45.34% on Qwen-2.5 1.5B, and 46.88% on GPT-2-Medium 355M. Every backbone is within 5 pp of the 42.5% random baseline, and 12 to 35 pp below the prior's `test_unseen` accuracy on the same checkpoints (Fig. 1); the like-for-like ACD comparison in §6 shows the same ordering. The generated text carries little concept signal: it degenerates into token-repetition loops or coherent but off-domain pretrain-mode text, and an LLM judge rates none of 20 CLS-INV samples as fluent reviews (Apps. F, G). **The collapse is invisible from training metrics**: they remain healthy to the final epoch, and the 50-step inversion optimiser *does* maximise classifier accuracy on $\mathbf{z}^\star$. The failure occurs at generation time.

The same collapse on a controlled sanity check. The collapse is not specific to Fyelp. On a synthetic 4-axis task with marginal-independent attributes, CLS-INV accuracy stays within ± 0.03 of the 0.25 baseline on five backbones across three families and 124M–3B parameters. On single-axis YelpP with LLaMA-3 8B, it remains at the 0.50 binary baseline (App. D, Tab. 5). The failure appears under both injector mechanisms — AdaLN-zero and the additive fallback for Qwen-1.5B — and is total on the joint all-axes-correct metric (≤ 0.005). This points to the inference protocol rather than a single backbone or injector design.

5.2 The mechanism: an off-manifold control code

The classifier-inversion objective (Eq. 1) contains no term that keeps $\mathbf{z}^\star$ within the code distribution on which the generator was trained. We show that the inverted code leaves this distribution, overdrives the injector, and that constraining it back toward the encoder manifold repairs the collapse.

The inverted code is measurably off-manifold. We fit a per-dimension diagonal Gaussian $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ to the encoder codes of 4000 training reviews, the code distribution on which the AdaLN injector was trained, and measure how far each inference-time $\mathbf{z}^\star$ lies from it (App. H). Classifier inversion is a clear outlier on both backbones: its Mahalanobis distance is 3.3–3.7, compared with 0.50–0.60 for the prior and ≈ 1.0 for REF-ENC. This gives a 3–7$\times$ gap, with the same separation in nearest-neighbour distance. The prior and REF-ENC lie inside the encoder's code distribution; the inverted code does not. This is measured directly, not inferred from downstream generation quality.

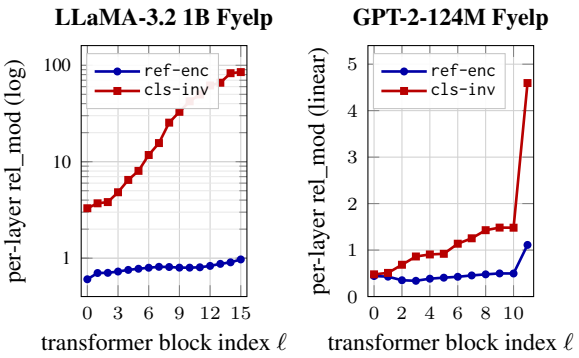

Figure 3: Activation-level signature of classifier inversion (§5.2). We plot per-layer relative modulation $\text{rel_mod}_\ell = \|\tilde{\mathbf{h}}_\ell - \mathbf{h}_\ell\| / \|\mathbf{h}_\ell\|$, averaged over tokens and prompts, on trained Fyelp checkpoints. REF-ENC remains bounded on both backbones, while CLS-INV produces depth-amplified over-modulation: severe on LLaMA-3.2 1B (log y-axis) and milder but still visible on GPT-2 124M. The corresponding off-manifold distances are reported in Table 10.

Off-manifold codes over-drive the injector. An off-manifold $\mathbf{z}^\star$ pushes the AdaLN injector outside its trained range. We probe per-layer relative modulation, $\text{rel_mod}_\ell = \|\tilde{\mathbf{h}}_\ell - \mathbf{h}_\ell\| / \|\mathbf{h}_\ell\|$, on the trained Fyelp checkpoints (Fig. 3). On LLaMA-3.2 1B, classifier inversion produces a depth-amplified perturbation: rel_mod rises from 3.3 at layer 0 to 85 at layer 15, averaging a **40**$\times$ over-shoot over REF-ENC. On GPT-2 124M, the same over-shoot is present but milder (2.7$\times$). Both backbones show the same signature: an off-manifold code amplified through depth, with a magnitude that depends on the backbone family. The resulting text degenerates into high-perplexity pretrain-mode loops (LLaMA-1B perplexity > 130).

Does a regulariser close the gap to the prior? A standard manifold regulariser repairs the bare-form collapse, and more targeted regularisers narrow the gap further. However, no tested variant matches the post-hoc prior. We re-run inversion with a label-agnostic manifold regulariser on

the per-axis objective, $\mathrm{CE}(\mathbf{W}_a\mathbf{z}_a, c_a^\star) + \beta R(\mathbf{z}_a)$, sweeping β under two penalty forms, two backbones, and three seeds (App. I). A *Mahalanobis* penalty, $R = \mathbb{E}[((\mathbf{z}_a - \boldsymbol{\mu}_a)/\boldsymbol{\sigma}_a)^2]$, lifts 4-axis accuracy from chance to 57.4% on GPT-2 and 52.8% on LLaMA, while reducing perplexity from ~ 130 to ~ 20. A *shell* penalty is zero inside the $\pm 2\sigma$ ellipsoid and has no centring pull, avoiding a trivial mean-collapse explanation. It still lifts accuracy by $+8.5$ and $+9.0$ pp; every lift exceeds seed noise (≤ 1.6 pp) by roughly an order of magnitude.

A manifold constraint, under either penalty and on either backbone, converts a chance-level protocol into a working one, supporting the off-manifold diagnosis. This is the failure mode that regularisation in classifier-gradient methods helps prevent. The best regularised inversion variant still trails the label prior by 7 to 29 pp on `test_seen`, as does a conditional normalising-flow density baseline (Tab. 2; Apps. I, K, J for an inversion-side hyperparameter grid). We therefore adopt the prior as the recommended protocol.

6 A label-conditioned prior recovers compositional control

If classifier inversion is an unreliable way to obtain a control code from a target configuration, what alternative inference protocol should be used? We show that a post-hoc label-conditioned prior, which estimates the conditional mean of the encoder code given the target labels, recovers compositional control on the same checkpoints where classifier inversion fails, with no inference-time optimisation and no reference text.

6.1 Prior inference recovers compositional generalisation

We evaluate the post-hoc label-conditioned MLP g_γ on CompMCTG Fyelp under the official RoBERTa-large 4-axis evaluator. We compare it against two reference protocols: the collapsed classifier-inversion protocol CLS-INV and the single-sample reference encoding REF-ENC.

Prior inference recovers control on every backbone. While classifier inversion remains near the $\approx 42.5\%$ random baseline even on `test_seen` (Tab. 1 (†)), the prior reaches 55.18% (GPT-2 124M), 61.50% (GPT-2-Medium 355M), 64.18% (Qwen-2.5 1.5B), and 76.10% (LLaMA-3.2 1B) 4-axis accuracy on the Hold-Out `test_unseen` split — a $+12$ to $+35$ pp recovery on identical checkpoints. The comparison is confirmed like-for-like on ACD, where half of all attribute combinations are unseen and both protocols can be read off the same split without the singleton artefact: on ACD `test_unseen`, the prior improves over CLS-INV by $+7.4$ to $+18.8$ pp (Tab. 1, within-split block; App. R). The gain is not merely over the bare protocol. Tab. 2 compares the prior with regularised inversion variants on matched checkpoints. Both a label-agnostic Mahalanobis penalty (App. I) and a sharper label-conditioned variant, which pulls $\mathbf{z}_a$ toward the per-(axis, target-label) marginal mean, still trail the prior by 7 to 29 pp. They also require a 50-step inner optimisation per query and a backbone-specific β sweep, both of which the deterministic prior avoids.

The prior also exceeds the encoded reference REF-ENC. On Fyelp Hold-Out, the prior improves over REF-ENC on every backbone by $+3.3$ to $+12.3$ pp in Tab. 1. The only exception is Qwen-1.5B on ACD, where the prior is -1.6 pp below REF-ENC. We do *not* interpret this as surpassing an upper bound: REF-ENC is a single-sample diagnostic and carries the surface idiosyncrasies of one reference sentence. A conditional-mean estimator can therefore outperform it by averaging away sample-specific variation, confirming that g_γ acts as the intended per-combination denoiser (§6.2). The same direction holds on the synthetic check (App. D), survives the bf16/fp32 recipe shift within ± 1.0 pp (App. P), and is reproduced by an LLM judge on both attribute match and fluency (App. G). The prior also beats a non-parametric nearest-seen-combination retrieval baseline on every unseen split by $+2.6$ to $+9.5$ pp (App. L).

6.2 Mechanism: prior as a conditional-mean denoiser

Why the conditional mean is the right target. Three properties of the two objectives account for the gap between them. *(i)* The inversion objective (Eq. 1) depends on $\mathbf{z}$ only through the per-axis classifier heads $\mathbf{W}_a$, so nothing in it penalises displacement from the code distribution on which the injector was trained; its minimisers are unconstrained on that distribution. *(ii)* The prior objective (Eq. 2) is a squared loss, so on the combinations it is fitted to its population minimiser is the conditional mean,

$$g_\gamma(\mathbf{c}) = \mathbb{E}[E_\phi(\mathbf{x}) \mid \mathbf{c}]\,, \qquad \mathbf{c} \in \mathcal{C}_{\text{seen}}, \quad (3)$$

Method / z-source	Backbone	HO	ACD
CompMCTG-paper baselines, GPT-2-Medium 355M (quoted from Zhong et al. 2024)			
PPLM (Dathathri et al., 2020)	GPT-2-M 355M	42.49	40.63
Fudge (Yang and Klein, 2021)	GPT-2-M 355M	41.71	40.42
CatPrompt (Yang et al., 2023)	GPT-2-M 355M	64.57	54.77
DCG (Zeng et al., 2023)	GPT-2-M 355M	66.39	64.71
Con.Prefix (Qian et al., 2022)	GPT-2-M 355M	67.50	63.93
CTRL (Keskar et al., 2019)	GPT-2-M 355M	68.29	65.31
Prior (Gu et al., 2023)	GPT-2-M 355M	54.64	52.29
Dis-Lens (Gu et al., 2022)	GPT-2-M 355M	67.06	57.31
Meta-CTRL (Zhong et al., 2024)	GPT-2-M 355M	**68.69**	**65.77**
Ours — 65K-train `_full` *runs, 4-axis mean accuracy (%)*			
CLS-INV†	GPT-2 124M	42.99	40.14
REF-ENC	GPT-2 124M	47.64	49.46
PRIOR	GPT-2 124M	**55.18**	**55.91**
CLS-INV†	GPT-2-M 355M	46.88	43.72
REF-ENC	GPT-2-M 355M	$49.79_{\pm 0.08}$	$46.73_{\pm 0.06}$
PRIOR	GPT-2-M 355M	$\mathbf{61.50}_{\pm 0.67}$	$\mathbf{59.52}_{\pm 0.32}$
CLS-INV†	LLaMA-3.2 1B	$41.47_{\pm 0.46}$	34.51
REF-ENC	LLaMA-3.2 1B	$63.80_{\pm 3.16}$	67.79
PRIOR	LLaMA-3.2 1B	$\mathbf{76.10}_{\pm 0.06}$	**69.53**
CLS-INV†	Qwen-2.5 1.5B	45.34	34.82
REF-ENC	Qwen-2.5 1.5B	60.90	62.04
PRIOR	Qwen-2.5 1.5B	**64.18**	**60.48**
Within-split sanity, ACD `test_unseen`‡ *(artefact-free; App. R for per-axis)*			
CLS-INV‡	GPT-2 124M / LLaMA-1B / Qwen-1.5B	—	48.47 / 50.75 / 52.82
REF-ENC	GPT-2 124M / LLaMA-1B / Qwen-1.5B	—	49.46 / 67.79 / 62.04
PRIOR	GPT-2 124M / LLaMA-1B / Qwen-1.5B	—	**55.91 / 69.53 / 60.48**

Table 1: 4-axis mean accuracy (%) on CompMCTG Fyelp under the official RoBERTa-large evaluator (Zhong et al., 2024); Hold-Out (HO) and ACD splits, idx=-0, chance baseline $\approx 42.5\%$. Unmarked rows report `test_unseen` (the compositional-generalisation number); † CLS-INV on `test_seen` (its Hold-Out `test_unseen` number is inflated by a singleton artefact, App. Q); ‡ artefact-free `test_unseen` on the ACD split, whose half-combination hold-out admits no singleton — the corresponding within-split lift over the PRIOR cells is $+7.4$ to $+18.8$ pp. Multi-seed entries (42/7/11) are marked $_{\pm \text{sd}}$. **Backbones differ**: every CompMCTG baseline uses GPT-2-Medium 355M; the strictly matched comparison is our GPT-2-Medium 355M block, where PRIOR does *not* surpass Meta-CTRL on either split; the LLaMA-1B win is cross-scale ($\approx 3.5\times$ the baseline backbone). Per-axis breakdown in Tab. 7 (App. E).

which averages away variation not shared across examples of the same combination and which we measure to lie inside the encoder's code distribution (§5.2). *(iii)* Whether that conditional mean extends to combinations never seen in training depends on how far the code factorises across axes, which we quantify below.

At training time, $E_\phi(\mathbf{x})$ encodes both per-combination concept information and sentence-specific stylistic variation. Fitted to the per-combination mean, g_γ therefore acts as a denoised prototype: it retains the shared code component of a combination and discards idiosyncratic sentence-level variation. Consistent with this view, prior PPL is lower than REF-ENC PPL across all three backbones.

The sample-specific noise is measurable and exceeds the concept signal. On the GPT-2-Medium checkpoint, the mean per-dimension σ of $E_\phi(\mathbf{x})$ within a fixed combination is 0.43, against only 0.33 across the per-combination means: REF-ENC carries the full 0.43 into the generator, whereas g_γ, the conditional mean, discards it and retains the 0.33 of concept signal. A direct control, averaging k same-combination reference encodings, matches the learned prior within 0.4 pp at $k = 256$ (App. M). Conversely, a conditional normalising flow $p(\mathbf{z} \mid \mathbf{c})$ underperforms the MLP by $10.5/11.8$ pp, consistent with the view that preserving within-combination variance is harmful for this inference protocol (App. K). A synthetic concept-swap diagnostic shows a compatible family-level pattern, with partial per-axis fidelity on LLaMA and stronger entanglement on GPT-2/Qwen; we report it only as a supplementary check (App. S).

Protocol	GPT-2-124M	LLaMA-1B
CLS-INV bare	42.99	41.47
+ Mahal. (agn.)$^{\diamond}$	57.52	52.91
+ Mahal. (cond.)$^{\diamond}$	60.74	52.73
`flow-prior` (cond. NF)†	57.21	69.95
PRIOR (g_γ)	**67.7**	**81.7**

Table 2: Inversion-protocol comparison on Fyelp Hold-Out `test_seen`, 4-axis mean accuracy (%); the same checkpoints report $55.18\%/76.10\%$ on `test_unseen` (Tab. 1). *agn./cond.*: label-agnostic / label-conditioned per-axis manifold penalty (App. I; best $\beta \in \{0.1, 0.3, 1.0, 3.0\}$, $^{\diamond}$; the label-agnostic row is the single-seed full `test_seen` number, matching the $57.4/52.8$ three-seed subset means in Tab. 11). `flow-prior`: conditional normalising flow $p(\mathbf{z} \mid c)$ (†; App. K); trails the MLP by 10.5/11.8 pp, matching the denoiser prediction (§6.2). Label-conditioned row uses the 312-row subset (Tab. 12); others use full `test_seen`. Same ordering on ACD (Tab. 1 within-split block).

Additivity of the concept code, and why the prior generalises. The prior can generalise to unseen combinations when the frozen encoder code $\mathbf{z}$ is sufficiently structured by attribute axes. An additive per-axis model $\boldsymbol{\mu} + \sum_a \Delta_a(y_a)$ explains only 31.7–47.6% of the variance of $\mathbf{z}$, increasing with backbone scale. However, the remaining non-additive residual is not explained by measurable label co-occurrence: all six Fyelp axis pairs have zero mutual information and Cramér's V, since CompMCTG samples combinations uniformly and the axes are independent by construction. We therefore interpret the residual primarily as a label-invariant encoder artefact dominated by per-sample variation, the same variation discarded by the conditional-mean prior. An additive label prior is therefore well matched to Fyelp, and adding an explicit interaction term gives no reliable gain on either backbone tested (App. N).

7 Baseline comparison and cross-benchmark robustness

The prior-over-classifier-inversion result of §6 is already evaluated under the official CompMCTG RoBERTa-large evaluator (§4). We now compare it against published CompMCTG baselines on Fyelp, test cross-dataset robustness on a second CompMCTG benchmark, Amazon, and add a single-axis YelpP check against CB-LLMs under their independently built RoBERTa evaluator.

7.1 Comparison with the CompMCTG baselines

We place the prior result on CompMCTG Fyelp (Tab. 1) against the CompMCTG-paper baselines (Zhong et al., 2024) under the same official evaluator.

Matched-backbone comparison: GPT-2-Medium 355M. The only strictly matched comparison retrains our framework at GPT-2-Medium 355M. In this setting, the prior reaches 61.50% on Hold-Out and 59.52% on ACD in 4-axis unseen accuracy over three training seeds, slightly below the strongest baseline. However, the prior sits above the mean of the nine published CompMCTG baselines on both splits (60.15% HO, 56.13% ACD), and above the median on ACD, where it beats PPLM, Fudge, Gu et al.'s Prior, CatPrompt, and Dis-Lens. **The contribution is an inference-protocol correction, not an accuracy record.** The matched setting nonetheless establishes the within-method ordering with seed-level error bars far smaller than the reported gaps: the prior beats REF-ENC by +12.8 pp on ACD unseen and collapsed classifier inversion by +14–16 pp, while achieving roughly 3× lower perplexity than Meta-CTRL. The 4-axis score is depressed roughly uniformly by the near-chance gender axis; the matched prior reaches 72.34% on 3-axis ACD unseen.

Cross-scale result and scaling trend. For completeness, on Fyelp Hold-Out our LLaMA-3.2 1B prior reaches 76.10% 4-axis unseen accuracy, +7.41 pp over Meta-CTRL. We label this as a cross-scale comparison rather than a state-of-the-art claim, since LLaMA-3.2 1B has roughly 3.5× the parameters of the GPT-2-Medium baselines. Prior Hold-Out accuracy increases with backbone size across our four backbones, a family-confounded trend showing only that the protocol does not saturate in the tested range. Only the LLaMA-1B point clears Meta-CTRL, and only cross-scale.

Second real-world dataset: Amazon. To check that the result is not Fyelp-specific, we run the same GPT-2-124M pipeline on CompMCTG **Amazon**, a product-review dataset with two axes, sentiment × topic, under the official 2-axis classifier suite (App. C). The ordering reproduces: on the seen split, PRIOR reaches 76.5%, beating collapsed classifier inversion by +30.0 pp and REF-ENC by +14.4 pp, and it is also best on the unseen split.

The collapse is clearest on the multi-class topic axis: classifier inversion is near chance (20.2% vs. 16.7%), whereas the prior reaches 74.3%, with a 5× perplexity gap. The binary sentiment axis partly survives, but the content-bearing topic axis collapses cleanly and the prior-over-REF-ENC advantage reproduces.

Single-axis check at a matched backbone. A minimal single-axis YelpP experiment against CB-LLMs (Sun et al., 2025) uses LLaMA-3 8B backbone and RoBERTa evaluator (App. B). The like-for-like result is consistent with the multi-axis findings: classifier inversion collapses to the 0.500 binary-chance level, while PRIOR reaches 0.964 steerability, +1.4 pp above the 0.95 reported by CB-LLMs under our byte-identical reproduction.

8 Conclusion

Classifier inversion silently collapses to chance on every tested backbone while training metrics remain healthy. We trace the failure to an off-manifold inference-time code that over-drives the injector, and confirm the diagnosis with a manifold-regulariser ablation. A post-hoc label-conditioned prior, fitted to per-combination encoder means, recovers compositional generalisation on the same checkpoints under external evaluators and on a second real-world dataset. At a matched backbone, it does not beat the strongest CompMCTG baseline; this is an inference-protocol correction, not a state-of-the-art claim. **For this family of concept-bottleneck CTG models, prior inference should be the default z-source rather than bare classifier inversion.**

Limitations

Benchmark granularity. The Fyelp Hold-Out idx=−0 split holds out a single 4-axis combination, so its `test_unseen` score is a coarse single-combination measurement and can inflate classifier-default artefacts (App. Q). We therefore also report ACD, where half of all combinations are unseen, as the more reliable compositional-shift measurement.

Architectural scope. Our collapse claim applies to concept-bottleneck CTG models whose generator is conditioned through an injector trained on the encoder's code distribution. This covers the CB-LLM-style architecture studied here, but not all possible controllable-generation mechanisms.

Mechanistic scope. The off-manifold diagnosis is supported by distance measurements, activation diagnostics, and manifold-regulariser ablations. However, it does not fully explain why the same off-manifold displacement yields different activation magnitudes across backbone families. A more detailed analysis of injector sensitivity, depth effects, and hidden-state scaling is left for future work.

Scale, language, and regulariser coverage. Experiments are limited to English benchmarks and to backbones up to 3B for multi-axis CTG, with an 8B single-axis check. Multilingual, long-form, and larger-scale multi-axis generation remain untested. Some backbone–dataset cells are single-seed due to compute cost, although the multi-seed blocks show protocol gaps larger than seed variation. Finally, we do not claim that no regularised inversion method can close the gap: we test label-agnostic and label-conditioned Mahalanobis penalties, a shell penalty, and a conditional normalising-flow density baseline, but broader regularisation schemes remain future work.

Ethical considerations

The Fyelp benchmark includes a binary *gender* attribute inherited from prior style-transfer datasets. We treat this label only as a benchmark-specific writing-style marker, not as an identity attribute, and do not advocate generating gendered content about identifiable people. Gender is used only through the standard CompMCTG evaluator; we do not train or deploy an additional gender classifier beyond the benchmark protocol.

This work diagnoses inference protocols on public benchmarks and releases no new dataset or deployed model. The label-conditioned prior improves the reliability of controlling already-labelled attributes, but does not expand the set of controllable attributes beyond those present in the training data. Misuse risks are therefore tied to the underlying controllable-generation setting rather than to a new capability introduced here.

Acknowledgements

We used AI assistants for language polishing, copy-editing, and minor coding/debugging assistance. All scientific claims, experimental design, analyses, and final text were reviewed and verified by the authors, who take full responsibility for the submission.

References

Milan Bhan, Yann Choho, Pierre Moreau, Jean-Noel Vittaut, Nicolas Chesneau, and Marie-Jeanne Lesot. 2025. Towards achieving concept completeness for textual concept bottleneck models. *Preprint*, arXiv:2502.11100.

Kumar Satvik Chaudhary, Chengshuai Zhao, Fan Zhang, Garima Agrawal, Yuli Deng, and Huan Liu. 2025. EssayCBM: Rubric-aligned concept bottleneck models for transparent essay grading. *Preprint*, arXiv:2512.20817.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations (ICLR)*.

Zijian Feng, Hanzhang Zhou, Zixiao Zhu, and Kezhi Mao. 2024. FreeCtrl: Constructing control centers with feedforward layers for learning-free controllable text generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, and Bing Qin. 2022. A distributional lens for multi-aspect controllable text generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, and Bing Qin. 2023. Controllable text generation via probability density estimation in the latent space. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning (ICML)*.

Brandon Hsu, Daniel Beaglehole, Adityanarayanan Radhakrishnan, and Mikhail Belkin. 2026. Contextual linear activation steering of language models. *Preprint*, arXiv:2604.24693.

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 (ICLR)*.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *Preprint*, arXiv:1909.05858.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations (ICLR)*.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *International Conference on Machine Learning (ICML)*.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP (Findings of EMNLP)*.

Vaibhav Kumar, Hana Koorehdavoudi, Masud Moshtaghi, Amita Misra, Ankit Chadha, and Emilio Ferrara. 2023. Controlled text generation with hidden representation transformations. In *Findings of the Association for Computational Linguistics (ACL Findings)*.

Roberto Labadie-Tamayo, Djordje Slijepčević, Xihui Chen, Adrian Jaques Böck, Andreas Babic, Liz Freimann, Christiane Atzmüller, and Matthias Zeppelzauer. 2025. Distilling knowledge from large language models: A concept bottleneck model for hate and counter speech recognition. *Information Processing & Management*.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. 2025. Programming refusal with conditional activation steering. In *International Conference on Learning Representations (ICLR)*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wendi Li, Wei Wei, Kaihe Xu, Wenfeng Xie, Dangyang Chen, and Yu Cheng. 2024. Reinforcement learning with token-level feedback for controllable text generation. In *Findings of the North American Chapter of the Association for Computational Linguistics (NAACL Findings)*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith,

and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li, and Zhiting Hu. 2023. Composable text controls in latent space with ODEs. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Narmeen Oozeer, Luke Marks, Shreyans Jain, Fazl Barez, and Amirali Abdullah. 2025. Beyond linear steering: Unified multi-attribute control for language models. In *Findings of the Association for Computational Linguistics: EMNLP (Findings of EMNLP)*.

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. 2018. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. In *Findings of the Association for Computational Linguistics (ACL Findings)*.

Tathagato Roy and Rahul Mishra. 2024. One arrow, many targets: Probing LLMs for multi-attribute controllable text summarization. *Preprint*, arXiv:2411.01213.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. 2025. Improving instruction-following in language models through activation steering. In *International Conference on Learning Representations (ICLR)*.

Chung-En Sun, Tuomas Oikarinen, Berk Ustun, and Tsui-Wei Weng. 2025. Concept bottleneck large language models. In *International Conference on Learning Representations (ICLR)*.

Chung-En Sun, Tuomas Oikarinen, and Tsui-Wei Weng. 2024. Crafting large language models for enhanced interpretability. In *ICML Workshop on Mechanistic Interpretability*.

Zhen Tan, Tianlong Chen, Zhenyu Zhang, and Huan Liu. 2024. Sparsity-guided holistic explanation for LLMs with interpretable inference-time intervention. In *AAAI Conference on Artificial Intelligence*.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Kexin Yang, Dayiheng Liu, Wenqiang Lei, Baosong Yang, Mingfeng Xue, Boxing Chen, and Jun Xie. 2023. Tailor: A soft-prompt-based approach to attribute-based controlled text generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Weihao Zeng, Lulu Zhao, Keqing He, Ruotong Geng, Jingang Wang, Wei Wu, and Weiran Xu. 2023. Seen to unseen: Exploring compositional generalization of multi-attribute controllable dialogue generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Tianqi Zhong, Zhaoyi Li, Quan Wang, Linqi Song, Ying Wei, Defu Lian, and Zhendong Mao. 2024. Benchmarking and improving compositional generalization of multi-aspect controllable text generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Appendix

A Training details

This appendix specifies the injector update, training objective, and loss schedule for the framework in §3.1.

Concept injector. Our default injector is AdaLN (Peebles and Xie, 2023). At every generator block ℓ, we apply

$$\tilde{\mathbf{h}}_\ell^{(t)} = (1 + \mathbf{s}_\ell(\mathbf{z})) \odot \mathbf{h}_\ell^{(t)} + \mathbf{b}_\ell(\mathbf{z}), \qquad (4)$$

where the scale and shift networks $\mathbf{s}_\ell, \mathbf{b}_\ell$ are zero-initialised. For Qwen-2.5 1.5B, AdaLN-zero was unstable in our runs, so we use an additive fallback:

$$\tilde{\mathbf{h}}_\ell^{(t)} = \mathbf{h}_\ell^{(t)} + s\,\sigma(g_\ell)\,P(\mathbf{z}), \qquad (5)$$

with gate initialisation $g_{\text{init}} = -3.0$ and front scale $s = 1$. We use a negative gate initialisation rather than reducing s, since the former only bounds the initial perturbation, whereas the latter also scales the gradient through the perturbation at every update step.

Training objective. With trainable parameters $\theta = (\phi, \psi, \theta_{\text{LoRA}})$, we minimise

$$\mathcal{L} = \mathcal{L}_{\text{gen}} + \lambda_{\text{c}}(\tau)\mathcal{L}_{\text{concept}} + \lambda_{\text{o}}(\tau)\mathcal{L}_{\text{orth}}, \qquad (6)$$

where $\mathcal{L}_{\text{gen}}$ is teacher-forced language-model cross-entropy using $\mathbf{z} = E_\phi(\mathbf{x})$, $\mathcal{L}_{\text{concept}} = \sum_a \text{CE}(\hat{\mathbf{p}}_a, c_a)$ supervises the encoder classifier heads, and

$$\mathcal{L}_{\text{orth}} = \mathbb{E}\left[\sum_{a \neq b} \left(\frac{\mathbf{z}_a^\top \mathbf{z}_b}{\|\mathbf{z}_a\|\,\|\mathbf{z}_b\|}\right)^2\right] \qquad (7)$$

penalises inter-axis cosine similarity. We use a four-phase schedule: first warming up $\mathcal{L}_{\text{gen}}$, then ramping λ_{c} and λ_{o} in sequence. The per-phase weights are listed in Tab. 17.

Why no intervention-consistency term. Earlier versions included an intervention-consistency loss: when the sub-vector for axis a is replaced, the classifier-head predictions for axes $b \neq a$ should remain unchanged. In our per-axis bottleneck, this term is identically zero. Each classifier head $\mathbf{W}_b$ reads only its own sub-vector $\mathbf{z}_b$, so replacing $\mathbf{z}_a$ for $a \neq b$ leaves $\mathbf{W}_b\mathbf{z}_b$ unchanged. The loss and its gradient are therefore zero by construction. We consequently use only the three-term objective in Eq. 6. The orthogonality regulariser remains active throughout phase 4 with weight 0.1 for all main-table checkpoints. All remaining constants were selected during preliminary validation on the synthetic 4-axis split and then held fixed across backbones.

B Single-axis YelpP check vs. CB-LLMs

To verify that the protocol findings are not artefacts of the multi-axis MCD setup, we run a deliberately minimal single-attribute experiment matched to the closest published concept-bottleneck generation work, CB-LLMs (Sun et al., 2025). CB-LLMs report a YelpP steerability score of **0.95** with a LLaMA-3 8B backbone under a RoBERTa-base classifier fine-tuned on `yelp_polarity` (their Table 5). We train our framework with the same LLaMA-3 8B base backbone, an AdaLN injector at every block, rank-8 LoRA, fp32-encoder mixed precision, and 25 epochs on the 6,000-sample YelpP single-axis split. We also re-run the `test_steerability.py` script from CB-LLMs on the official `cesun/cbllm-generation` checkpoint, exactly reproducing their reported score of **0.950** ($\Delta = 0.000$). The comparison is therefore head-to-head in backbone, evaluator, and metric.

Result. Our PRIOR reaches 0.964/0.964 (`test_seen` / `test_unseen`, $n = 6{,}000$ each) under the CB-LLMs RoBERTa evaluator, +1.4 pp above the reproduced CB-LLMs score (Tab. 3). Unlike the cross-scale CompMCTG comparison, this check is strictly backbone-matched, using LLaMA-3 8B on both sides. We therefore read the result as like-for-like corroboration of the protocol finding, not as a new state-of-the-art claim. The within-method PRIOR−REF-ENC gap is +6.9 pp, consistent with the conditional-mean denoising mechanism of §6.2.

Classifier inversion collapses to exactly 0.500 under our internal DistilBERT-SST-2 evaluator. Under the CB-LLMs RoBERTa evaluator, it instead scores 0.916, but the per-class breakdown (negative 0.83, positive 1.00) shows this to be a default-positive evaluator artefact on lexically degraded generations. The collapse is further confirmed by perplexity: CLS-INV has PPL 432, compared with 17 for REF-ENC, a 25× increase at the LLaMA-3 8B scale.

z-source	DistilBERT-SST-2	CB-LLMs RoBERTa
REF-ENC	0.846 / 0.850	0.895 / 0.891
PRIOR	**0.915 / 0.921**	**0.964 / 0.964**
CLS-INV	0.500 / 0.500	0.916 / 0.918
CB-LLMs (official ckpt, BOS)	—	**0.950**

Table 3: LLaMA-3 8B YelpP single-axis check (seen / unseen). Our PRIOR reaches 0.964 under the CB-LLMs RoBERTa evaluator, +1.4 pp above the reproduced CB-LLMs score at matched backbone and evaluator. YelpP is single-axis, so `test_seen` and `test_unseen` are random subsamples of the same distribution and agree within $\leq$ 0.7 pp. The high CB-LLMs RoBERTa score for CLS-INV reflects a default-positive prediction artefact on lexically degraded generations; the DistilBERT score and perplexity diagnostics show the collapse.

C Second real benchmark: CompMCTG Amazon

To test whether the prior-over-classifier-inversion result is specific to Fyelp, we run the same GPT-2-124M pipeline on a second CompMCTG dataset, **Amazon**. Amazon is a product-review benchmark with two attribute axes: sentiment with 2 values and topic with 6 values, giving 12 combinations. We use Hold-Out idx=-0, where the held-out combination is `positive_clothing`. The training recipe, three z-sources, and official CompMCTG evaluation protocol are unchanged from the Fyelp runs; only the dataset and 2-axis classifier suite differ.

z-source	split	sent.	topic	2-axis
CLS-INV†	seen	72.8	20.2	46.5
REF-ENC	seen	71.6	52.7	62.1
PRIOR	seen	**78.7**	**74.3**	**76.5**
CLS-INV†	unseen	99.4	9.7	54.6
REF-ENC	unseen	78.4	27.8	53.1
PRIOR	unseen	**94.9**	**61.2**	**78.1**

Table 4: CompMCTG Amazon results with GPT-2-124M under Hold-Out idx=-0, evaluated by the official 2-axis Amazon classifier suite. We report per-axis and 2-axis-mean accuracy (%). Topic is six-way classification, with 16.7% chance accuracy; sentiment is binary, with 50% chance accuracy. PRIOR is the best protocol on both splits. Classifier inversion collapses most clearly on the topic axis and has a 5× perplexity inflation on the seen split (PPL 65.1 vs. 13.0 for PRIOR). † On the unseen split, the held-out combination is a singleton with positive sentiment; the high CLS-INV sentiment score is therefore a classifier-default artefact (App. Q).

The protocol ordering reproduces. As shown in Tab. 4, PRIOR is the best protocol on both seen and unseen splits. On the seen split, it reaches 76.5% 2-axis accuracy, improving over classifier inversion by +30.0 pp and over REF-ENC by +14.4 pp. The latter gap reproduces the conditional-mean denoising advantage of §6.2 on a second real benchmark and a new domain.

The collapse concentrates on the content-bearing axis. Classifier inversion is near chance on the six-way topic axis: 20.2% on seen and 9.7% on unseen, compared with 74.3% and 61.2% for PRIOR. The binary sentiment signal partly survives in degraded text, so the 2-axis mean collapse is less complete than the 4-axis Fyelp collapse. Nevertheless, the content-bearing topic axis collapses cleanly, and the elevated classifier-inversion perplexity (65.1 vs. 13.0) matches the off-manifold signature documented on Fyelp (§5).

D Synthetic 4-axis sanity check

The synthetic 4-axis MCD task (§4) has marginally independent attributes and a controlled compositional split. We use it only as a sanity check for the two main protocol findings: classifier inversion collapses to the random baseline (§5), and the prior improves over single-sample reference encoding (§6). The synthetic task is not load-bearing for the main natural-language claims, which are supported by Fyelp, Amazon, and YelpP.

The prior-over-REF-ENC direction also reproduces on the synthetic task (Tab. 6): the prior improves over REF-ENC on every backbone, consistent with the denoising pattern observed on natural data.

E Per-axis Fyelp Hold-Out breakdown

Tab. 1 reports the 4-axis mean; this appendix gives the per-axis Hold-Out `test_unseen` accuracy behind it in Tab. 7.

F Qualitative generation samples

Tab. 8 shows representative generations for one target configuration on the GPT-2-124M Fyelp Hold-Out checkpoint, one per z-source. The example

Backbone	Scale	Task	CLS-INV mean
GPT-2	124M	synthetic 4-axis	**0.239**
Qwen-0.5B	500M	synthetic 4-axis	**0.276**
Qwen-1.5B	1.5B	synthetic 4-axis	**0.239**
LLaMA-1B	1B	synthetic 4-axis	**0.280**
LLaMA-3B	3B	synthetic 4-axis	**0.224**
LLaMA-3 8B	8B	YelpP binary†	**0.500**

Table 5: Classifier-inversion collapse on the synthetic 4-axis sanity check and single-axis YelpP. CLS-INV mean accuracy on generated text stays within ± 0.03 of the random baseline on every backbone. †YelpP is single-axis binary, with random baseline 0.50; the synthetic rows are four-axis, with random baseline 0.25.

Backbone	CLS-INV	REF-ENC	**PRIOR**	Δ pp
GPT-2 124M	0.239	0.536	**0.574**	$+3.8$
Qwen-0.5B	0.276	0.603	**0.679**	$+7.55$
Qwen-1.5B†	0.239	0.360	**0.379**	$+1.9$
LLaMA-1B (n=5)	0.280	**0.791** $\pm$ **0.021**	**0.840** $\pm$ **0.026**	$+4.86 \pm 0.96$
LLaMA-3B	0.224	0.857	**0.907**	$+4.93$

Table 6: Synthetic 4-axis unseen mean accuracy for classifier inversion, reference encoding, and the post-hoc prior. The prior improves over REF-ENC on every backbone, matching the direction observed on natural data (§6). † Qwen-1.5B uses the additive injector with gate initialisation -3.0. The LLaMA-1B row reports the 5-seed mean $\pm$ standard deviation.

makes the collapse of §5 visible at the surface level: classifier inversion produces a degenerate token-repetition loop, whereas REF-ENC and PRIOR produce fluent restaurant-review text. Samples are verbatim, lowercased as in the corpus, and truncated to their first ~30 words.

G LLM-as-judge qualitative evaluation

The headline numbers rest on automatic metrics (official classifier accuracy, perplexity, Dist-n). Because one of those metrics — the official classifier — has a documented default-prediction artefact on degraded text (App. Q), we add a small LLM-as-judge study as an independent qualitative check.

Protocol. From the GPT-2-124M Fyelp Hold-Out `test_seen` generations we drew 20 target combinations (seed 42) and, for each, the generation under every z-source — 60 texts, combination-matched across z-sources. An LLM judge (Claude Opus 4.7) rated each text under a fixed rubric, applied per generation. The rubric has two parts. **(a) Attribute match**: for each of the three policy-controlled axes (sentiment, cuisine, tense; gender excluded per the ethics scope of §8), the judge scores 1 if the text exhibits the target value for that axis and 0 otherwise — sentiment by the review's overall polarity, cuisine by the food/venue named, tense by the dominant verb tense — summed to a 0–3 attribute-match score. **(b) Fluency**: a single categorical label — *fluent* (a coherent, on-domain review), *degraded* (grammatical but incoherent or off-domain), or *token-salad* (repetition loop or word/character soup). Under uniform random guessing the per-axis hit rates are 0.5 (sentiment, binary), 0.2 (cuisine, 5-way), and 0.5 (tense, binary), so chance scores ≈ 1.2 on attribute match.

Findings (Tab. 9). (1) The prior satisfies the attributes: it scores $2.45/3$ on attribute match, against 1.90 for REF-ENC and 0.80 for CLS-INV — the last *below* the $\approx 1.2/3$ chance baseline, because collapsed text often expresses no attribute at all. **(2) The prior does not sacrifice naturalness:** all 20 prior generations are rated fluent, slightly ahead of REF-ENC ($16/20$) — the conditional-mean code does not trade fluency for control. **(3) The single-sample idiosyncrasy of REF-ENC is visible:** REF-ENC trails the prior on attribute match (-0.55) and fluency ($4/20$ degraded vs. 0), because an individual reference encoding carries the quirks of the chosen sentence — wrong tense, off-topic drift — exactly the sample-specific noise that the conditional mean removes (§6.2). **(4) Classifier-inversion failure is not only repetition:** no CLS-INV generation is a fluent review; $9/20$ are token-salad (repetition loops, word or character soup) and $11/20$ are degraded — grammatical but off-domain *pretrain-mode* text (city news, TV-show, essay fragments) carrying no attribute signal. The

Method / z-source	Backbone	Acc_s	Acc_g	Acc_c	Acc_t
CompMCTG-paper baselines (GPT-2-Medium 355M)					
PPLM	GPT-2-M	49.96	50.02	19.93	50.06
Fudge	GPT-2-M	49.61	48.80	20.91	47.50
CatPrompt	GPT-2-M	83.82	54.07	56.04	64.36
DCG	GPT-2-M	90.29	56.39	57.00	61.88
Con.Prefix	GPT-2-M	93.66	59.24	48.30	68.78
CTRL	GPT-2-M	87.88	59.65	59.02	66.61
Prior	GPT-2-M	63.56	50.79	43.58	60.62
Dis-Lens	GPT-2-M	77.03	56.05	78.23	56.93
Ours, Hold-Out idx=−0, 65K-train `_full` *runs, single seed*					
CLS-INV	GPT-2 124M	50.6	41.2	19.2	61.0
REF-ENC	GPT-2 124M	61.6	35.1	36.9	57.0
PRIOR	GPT-2 124M	66.3	39.1	52.1	63.2
CLS-INV	GPT-2-M 355M	55.4	63.6	19.8	48.7
REF-ENC	GPT-2-M 355M	52.9	41.4	39.8	64.9
PRIOR	GPT-2-M 355M	80.9	41.3	53.2	66.9
CLS-INV	LLaMA-3.2 1B	46.2	48.3	23.5	46.6
REF-ENC	LLaMA-3.2 1B	87.5	62.3	50.6	62.3
PRIOR	LLaMA-3.2 1B	98.1	67.6	64.9	75.0
CLS-INV	Qwen-2.5 1.5B	59.0	54.5	20.6	47.3
REF-ENC	Qwen-2.5 1.5B	80.6	62.8	38.2	61.9
PRIOR	Qwen-2.5 1.5B	83.8	60.2	50.9	61.9

Table 7: Per-axis CompMCTG Fyelp Hold-Out idx=−0 `test_unseen` accuracy (sentiment / gender / cuisine / tense), official RoBERTa-large evaluator; our rows are 65K-train `_full` runs, single seed (seed 42). Random baselines: s/g/t = 50.0, c = 20.0. Averaging the per-axis cells recovers the 4-axis means of Tab. 1, except its LLaMA-3.2 1B Hold-Out column: that column is a 3-seed mean whereas these per-axis rows are the single seed-42 run, so they differ — by 0.3 pp for PRIOR and CLS-INV but 1.9 pp for REF-ENC, whose single-combination unseen split is the most seed-sensitive. CLS-INV rows are reported on `test_seen` (per Tab. 1, footnote †). Gender is the weakest-controlled axis for almost every method — near or below the 50% chance level for the baselines and for classifier inversion — consistent with gender being barely encoded in $\mathbf{z}$ (§6.2).

collapse is therefore both degenerate looping and reversion to the pretraining prior of the backbone, both consistent with an off-manifold control code (§5.2).

H Off-manifold distance of the inverted code

Tab. 10 reports the direct measurement behind §5.2: the distance of each inference-time control code $\mathbf{z}^\star$ to the training-data code distribution of the encoder, on the Fyelp Hold-Out checkpoints.

I Manifold-regulariser sweep

Tab. 11 gives the full β sweep behind the regulariser ablation in §5: classifier inversion re-run with a label-agnostic manifold regulariser on the per-axis objective, using two penalty forms, two backbones, and three seeds.

Full-test-set regularised inversion. Tab. 11 sweeps β on a stratified 312-row subset; we also ran the best Mahalanobis setting on the *full* Hold-Out test sets. The lift holds: 4-axis seen accuracy rises from 42.99 to 57.52 on GPT-2-124M and from 41.47 to 52.91 on LLaMA-3.2 1B, matching the subset sweep. The regularised optimum still trails the label prior — by 10.2 and 28.8 pp on the seen split and 4.5 and 23.5 pp on the unseen split. A sharpened *label-conditioned* variant that pulls toward the per-(axis, target-label) marginal mean (Eq. 8, Tab. 12) modestly improves over the label-agnostic centroid on GPT-2 (+3.2 pp) but is tied on LLaMA, still trailing g_γ by 7 and 29 pp — the advantage of the prior is structural (per-combination, not per-axis), not merely optimisation cost.

Perplexity and diversity along the sweep. On the GPT-2-124M Hold-Out subset the regulariser collapses perplexity from 131.6 ($\beta = 0$, bare classifier inversion) to 18–23 for every $\beta > 0$, while 4-axis accuracy peaks at 58.7% ($\beta = 0.1$) and then declines. Distinct-2 is *highest* at $\beta = 0$ (0.50) and falls to $\sim$0.30 as β grows: the elevated diversity of bare classifier inversion is the token-salad signature (random n-grams), not useful variety, and must be read alongside the 131.6 perplexity.

Label-conditioned variant. A natural sharpening of the label-agnostic penalty in Tab. 11 replaces

z-source	Generated text (target: `negative_female_bar_past`)
CLS-INV	*class class class Class Class Class Kids Class Class Sing Kid Class Kids Special Kids Special Children Class Kids Kids Special Special Kids Children . . .*
REF-ENC	adios everyone, this place is totally worth going back to. the owner, who was seated at the bar, told us he had come to check out and that he had some frien . . .
PRIOR	was so disappointed in this place. i ordered the last meal i had, which was a very small portion of the pizza. the waiter was so nice and attentive . . .

Table 8: Representative generations per z-source on GPT-2-124M Fyelp Hold-Out for target `negative_female_bar_past`. Classifier inversion degenerates into a repetition loop, while REF-ENC and PRIOR produce fluent restaurant-review text; only PRIOR clearly reflects the negative target in this example.

z-source	attr. match (mean /3)	fluency ($n = 20$) fluent	degraded / salad
CLS-INV	0.80	0	11 / 9
REF-ENC	1.90	16	4 / 0
PRIOR	**2.45**	**20**	0 / 0

Table 9: LLM-as-judge evaluation: 20 combination-matched generations per z-source, GPT-2-124M Fyelp Hold-Out `test_seen`, judge = Claude under a fixed rubric. Attribute match counts the three policy-controlled axes; chance guessing $\approx 1.2/3$. The prior leads on both attribute match and fluency, and classifier inversion yields no fluent review.

the per-axis global mean $\boldsymbol{\mu}_a$ with the per-(axis, target-label) *conditional* mean, conditioned on the inference target $c_a^\star$:

$$R_{\text{cond}}(\mathbf{z}_a) = \text{mean}\Big(\big((\mathbf{z}_a - \boldsymbol{\mu}_a^{c_a^\star})/\boldsymbol{\sigma}_a^{c_a^\star}\big)^2\Big), \quad (8)$$

where $(\boldsymbol{\mu}_a^v, \boldsymbol{\sigma}_a^v)$ are the per-axis mean/std of the encoder code over training samples whose axis-a label is v. Conditioning the pull on the target label is a strict sharpening of the label-agnostic centroid and the natural candidate to close the gap to the prior. Tab. 12 reports the same β sweep as Tab. 11 on the same 312-row stratified subset, single seed (seed 42).

Why label-conditioning cannot close the gap. The label-agnostic penalty imposes only a manifold constraint — "stay near the training distribution of the encoder" — and leaves the directional choice to the cross-entropy loss against $\mathbf{W}_a$. The label-conditioned penalty also specifies *where on the manifold to head*: toward the per-axis marginal mean $\boldsymbol{\mu}_a^{c_a^\star}$. This direction averages over all training combinations that share $c_a^\star$ on axis a, regardless of the labels of the other axes. The post-hoc prior g_γ avoids this averaging by amortising *per-combination* means directly; that structural distinction (per-axis marginal → per-combination joint), not merely per-query optimisation cost, is why the prior beats every regularised inversion variant we tested.

Backbone	$\mathbf{z}^\star$ source	Mahalanobis	kNN-dist
GPT-2 124M	CLS-INV	$3.34_{\pm 0.06}$	$17.6_{\pm 0.5}$
	REF-ENC	$1.00_{\pm 0.21}$	$4.5_{\pm 0.5}$
	PRIOR	$\mathbf{0.50}_{\pm 0.05}$	$\mathbf{3.2}_{\pm 0.1}$
LLaMA-3.2 1B	CLS-INV	$3.74_{\pm 0.15}$	$10.2_{\pm 0.9}$
	REF-ENC	$1.00_{\pm 0.20}$	$2.4_{\pm 0.2}$
	PRIOR	$\mathbf{0.60}_{\pm 0.09}$	$\mathbf{1.7}_{\pm 0.1}$

Table 10: Distance of each inference-time control code to the encoder's training-data code distribution (Fyelp Hold-Out; mean$_{\pm \text{sd}}$ across the 39 seen combinations). Diagonal-Gaussian Mahalanobis and mean distance to 10 nearest training neighbours. Classifier inversion lands 3–7× farther off-manifold than the prior or an encoded reference, on both backbones, with the gap many times its combination-level spread.

	GPT-2-124M		LLaMA-3.2 1B	
β	Mahal.	shell	Mahal.	shell
0	$43.8_{\pm 0.6}$	$43.8_{\pm 0.6}$	$41.7_{\pm 1.2}$	$40.5_{\pm 0.5}$
0.03	$55.7_{\pm 1.3}$	$51.5_{\pm 0.6}$	$49.8_{\pm 0.3}$	$43.9_{\pm 0.8}$
0.1	$\mathbf{57.4}_{\pm 1.2}$	$\mathbf{52.2}_{\pm 0.4}$	$47.5_{\pm 0.8}$	$43.0_{\pm 0.7}$
0.3	$54.0_{\pm 0.8}$	$51.7_{\pm 1.3}$	$\mathbf{52.8}_{\pm 0.4}$	$49.1_{\pm 0.1}$
1	$50.5_{\pm 0.3}$	$50.7_{\pm 1.6}$	$48.5_{\pm 1.2}$	$\mathbf{49.6}_{\pm 0.8}$
3	$48.7_{\pm 1.1}$	$50.8_{\pm 1.5}$	$48.1_{\pm 1.0}$	$52.0_{\pm 5.1}$

Table 11: Classifier inversion with a label-agnostic manifold regulariser of weight β: 4-axis seen accuracy (%), mean$_{\pm \text{sd}}$ over 3 seeds (42/7/11). Fyelp, official evaluator, fixed 312-row stratified `test_seen` subset (8 reviews per seen combination); $\beta = 0$ reproduces plain classifier inversion. *Mahal.*: $((\mathbf{z}-\boldsymbol{\mu})/\boldsymbol{\sigma})^2$, pulls toward the per-axis centroid. *shell*: $\text{relu}(((\mathbf{z}-\boldsymbol{\mu})/\boldsymbol{\sigma})^2 - \rho^2)$, $\rho = 2$ — zero inside the $\pm 2\sigma$ ellipsoid, no centring pull. Every $\beta > 0$ lifts accuracy off chance on both backbones and both penalties (bold: per-column peak); the $+8$ to $+14$ pp lift exceeds the seed noise (≤ 1.6 pp on $23/24$ cells) by roughly $10\times$.

J Bare cls-inv: inversion-side hyperparameter grid

The bare-inversion collapse in §5 is measured at a single inversion-side configuration (50 Adam

β	GPT-2-124M	LLaMA-3.2 1B
0.1	**60.74**	**52.73**
0.3	56.65	48.31
1.0	53.93	49.92
3.0	54.89	49.36
Reference (from Tab. 11, 3-seed mean):		
Mahal., best β (label-agnostic)	57.4	52.8
prior g_γ (§6.2)	$\sim$67.7	$\sim$81.7

Table 12: Label-conditioned manifold regulariser (Eq. 8), 4-axis seen accuracy (%) on the same 312-row stratified Fyelp Hold-Out `test_seen` subset as Tab. 11, single seed (42). Bold: per-column peak. The label-conditioned penalty modestly outperforms the label-agnostic centroid on GPT-2-124M ($+3.2$ pp at $\beta = 0.1$) and roughly ties on LLaMA-3.2 1B (-0.1 pp), yet remains 7 and 29 pp below g_γ. The per-axis conditional marginal collapses the cross-axis structure that g_γ amortises directly per combination, so it cannot reach the prior's per-combination joint pull target.

steps, lr $= 0.1$, initialised from $\mathbf{W}_a[c_a^\star, :]^\top$). To rule out the alternative explanation that this single setting is merely *under-optimised*, we sweep $(\text{lr}, \text{steps}) \in \{0.01, 0.1, 1.0\} \times \{50, 200, 1000\}$ on the same 312-row Fyelp Hold-Out `test_seen` subset used in App. I, single seed (42), no regulariser ($\beta = 0$), with all other settings identical to the main run (Tab. 13). Chance is 42.5% (uniform over $4{\times}2{\times}5{\times}3$ axis labels with the per-axis size mix the evaluator uses).

	steps		
lr	50	200	1000
GPT-2-124M (paper default in bold):			
0.01	42.53	43.57	42.48
0.1	**43.63**	41.91	42.28
1.0	46.61	45.74	46.96
LLaMA-3.2 1B at steps$=1000$*:*			
0.01	—	—	44.23
0.1	—	—	41.74
1.0	—	—	45.75

Table 13: Bare CLS-INV accuracy (%) on a 3×3 inversion-side grid (GPT-2-124M) plus three LLaMA-3.2 1B sanity cells, Fyelp Hold-Out `test_seen` 4-axis mean. All 12 cells lie within ± 5 pp of the 42.5% chance line; the best cell (lr $= 1.0$, GPT-2) reaches 46.96%, still 10.5 pp below the cheapest manifold regulariser (Mahal. label-agn., 57.52%, Tab. 11) and 20.7 pp below the post-hoc prior. The bare-inversion collapse is therefore not an under-optimisation artefact of the paper's default schedule.

K Density-modelling baseline: conditional normalising flow

The post-hoc prior g_γ in the main text is a deterministic MLP fitted to the per-combination encoder *means*. The natural sharper alternative — and the closest competitor in the latent-CTG literature (Gu et al., 2023) — is a learned *conditional density* $p(\mathbf{z} \mid c)$. If the within-combination encoder variance carries useful signal, modelling the full conditional distribution should outperform predicting only its mean. The denoiser interpretation in §6.2 predicts the opposite: removing within-combination noise is what the deterministic mean delivers, so a density model should *underperform* the MLP.

We fit an 8-layer conditional RealNVP $p(\mathbf{z} \mid c)$ on the frozen encoder's training $(\mathbf{z}, c)$ pairs (per-combination one-hot labels), max-likelihood for 5000 Adam steps at lr $= 10^{-3}$, 128-unit affine-coupling MLPs. At inference, we sample $\mathbf{z} \sim p(\mathbf{z} \mid c^\star)$ given the target labels. The flow uses the same frozen encoder, the same generator checkpoint, and the same external CompMCTG evaluator as every other row of Tab. 2.

Result. On Fyelp Hold-Out `test_seen`, the flow-prior reaches 57.21% (GPT-2-124M) and 69.95% (LLaMA-3.2 1B) on the official 4-axis evaluator — a genuine working baseline, well above bare CLS-INV ($+14.2$/$+28.5$ pp) and above the label-agnostic Mahalanobis regulariser on LLaMA ($+17$ pp). It still trails the deterministic MLP g_γ by $\mathbf{10.5}$ pp (GPT-2) and $\mathbf{11.8}$ pp (LLaMA-1B). The ordering is consistent with the denoiser hypothesis: the density target $p(\mathbf{z} \mid c^\star)$ retains the within-combination variance that the conditional-mean estimator $\mathbb{E}[\mathbf{z} \mid c^\star]$ averages out, so sampling from the flow injects sample-specific variation that the injector of the generator then has to process. The implicit denoising of g_γ is therefore not a side-effect of architectural simplicity — a strictly more expressive density model with the same fitting data

underperforms by roughly 11 pp on each backbone.

L Nearest-seen-combo retrieval baseline

To check whether the learned prior g_γ does more than retrieve the closest observed combination, we compare it with a non-parametric baseline. Given a target combination, the baseline returns the per-combination mean encoder code of the nearest seen combination, using Hamming distance in label-index space; a seen target retrieves itself. Both methods are evaluated under the official CompMCTG evaluator on the four Fyelp checkpoints (Tab. 14).

	seen		unseen	
Run	retr.	prior	retr.	prior
GPT-2 124M, HO	68.8	67.7	46.6	**55.2**
GPT-2 124M, ACD	74.6	74.2	53.3	**55.9**
LLaMA-3.2 1B, HO	83.0	81.7	66.9	**76.4**
LLaMA-3.2 1B, ACD	87.0	86.9	64.0	**69.5**

Table 14: Nearest-seen-combination retrieval baseline versus the learned label prior g_γ: 4-axis accuracy (%) under the official CompMCTG evaluator. On seen combinations, retrieval and the prior agree within 1.3 pp, as both recover the observed per-combination mean. On every unseen split, the learned prior wins by $+2.6$ to $+9.5$ pp.

On seen combinations, retrieval and the learned prior are nearly equivalent, as expected. On unseen combinations, however, the prior wins consistently. This shows that g_γ is not merely a lookup table: its learned label-to-code map performs useful compositional interpolation where compositional generalisation is tested.

M Isolating within-combination denoising: the k-sample control

§6.2 attributes the advantage of the prior over a single-sample reference encoding to conditional-mean denoising. We test that directly with a non-parametric control. For each `test_seen` target combination we draw k training references *of that same combination*, encode each, and average their concept codes; the averaged code drives generation in place of $g_\gamma(\mathbf{c}^\star)$. As k grows this is an increasingly low-variance estimate of the per-combination mean $\mathbb{E}[E_\phi(\mathbf{x}) \mid \mathbf{c}]$ — exactly the quantity that Eq. 2 fits. The control is defined on `test_seen` only: unseen combinations have no training references. We sweep $k \in \{1, 4, 16, 64\}$ with three independent reference draws each, plus a single k=256 draw, on GPT-2-124M for both the Hold-Out and ACD Fyelp checkpoints, scored by the official 4-axis evaluator. The k=1 row here is one training-pool reference scored on `test_seen`; it is therefore not the REF-ENC row of Tab. 1, which encodes each test sentence's own text and is scored on `test_unseen`.

k references	Hold-Out	ACD
1	$61.52_{\pm 2.00}$	$67.62_{\pm 0.74}$
4	$62.70_{\pm 0.31}$	$69.38_{\pm 0.43}$
16	$63.35_{\pm 0.31}$	$70.60_{\pm 0.26}$
64	$63.65_{\pm 0.26}$	$70.07_{\pm 0.16}$
256	68.05	74.45
prior g_γ	67.67	74.23
retrieval ($k \to \infty$)	68.8	74.6

Table 15: The k-sample within-combination averaging control: 4-axis `test_seen` accuracy (%) under the official evaluator, mean$_{\pm \text{sd}}$ over three reference draws for $k \leq 64$. Averaging more same-combination references denoises the code: accuracy rises with k and the cross-draw spread collapses. By k=256 the non-parametric average reaches the learned prior g_γ to within 0.4 pp on both datasets.

Three things hold on both datasets (Tab. 15). **(i)** Accuracy increases with k — a single reference is a noisy code, and averaging denoises it. **(ii)** The cross-draw standard deviation collapses, from ± 2.0 at k=1 to below ± 0.3 for $k \geq 64$ — the $\propto 1/\sqrt{k}$ signature of a conditional-mean estimator. **(iii)** By k=256 the average reaches 68.05 (Hold-Out) and 74.45 (ACD), within 0.4 pp of the learned prior g_γ. In the limit $k \to \infty$ the within-combination average is, by construction, the full per-combination training mean; we verified numerically that this mean is byte-identical (cosine 1.000) to the nearest-seen-combo code of the retrieval baseline (App. L), whose accuracy (68.8 / 74.6) brackets g_γ. The edge of the prior over a single-sample REF-ENC is thus confirmed to be within-combination denoising: averaging enough references of a combination reproduces the accuracy of the prior.

N Concept-code additivity and the factorised prior

This appendix supports the additivity analysis in §6. We ask three questions: how additive the frozen-encoder concept code is, whether the non-additive residual is explained by label co-occurrence, and whether adding an explicit interaction term improves the label prior.

	sentiment	gender	cuisine	tense
sentiment	–	0.00	0.00	0.00
gender	0.00	–	0.00	0.00
cuisine	0.00	0.00	–	0.00
tense	0.00	0.00	0.00	–

Figure 4: Inter-attribute statistical dependence on the full Fyelp corpus (70K records, all 40 attribute combinations). Each off-diagonal cell is the normalised mutual information (NMI $\in [0, 1]$) between two attribute axes. **Every pair is exactly** 0.00 — raw mutual information and Cramér's V are likewise exactly zero, and every value-pair pointwise mutual information is $+0.000$. The four Fyelp axes are *independent by construction*: CompMCTG samples attribute combinations uniformly. The non-additive residual of the concept code (§6.2) therefore cannot be attribute interaction — it is an encoder artefact.

The concept code is only partially additive. We fit an additive per-axis model to the frozen-encoder code,

$$\hat{\mathbf{z}} = \boldsymbol{\mu} + \sum_{a=1}^{A} \Delta_a(y_a), \qquad \Delta_a(v) = \mathbb{E}[\mathbf{z} \mid y_a = v] - \boldsymbol{\mu}. \tag{9}$$

and report the fraction of variance explained on the Fyelp training pool (Tab. 16). The additive model explains 31.7% of the variance at GPT-2 124M, 38.3% at GPT-2-Medium 355M, and 47.6% at LLaMA-3.2 1B. Thus additivity increases with backbone scale, but even the largest backbone leaves more than half of the concept-code variance unexplained by per-axis offsets. The per-block breakdown is also informative: the gender block is weakly axis-localised at every scale (5–12%), consistent with gender being the weakest attribute signal in the CompMCTG Fyelp results.

The residual is not explained by label co-occurrence. A natural interpretation of the non-additive residual is that the attributes interact, e.g., that the code for *negative* × *Mexican* is not a linear superposition of its parts. We therefore measure statistical dependence between the four Fyelp axes on the full corpus. All six axis pairs have zero mutual information, normalised mutual information, and Cramér's V; value-pair PMI is also $+0.000$ throughout (Fig. 4). This confirms that Fyelp does not contain measurable label co-occurrence structure for the prior to exploit. The remaining non-additivity is therefore better interpreted as arising from the learned encoder representation, rather than from dataset-level attribute dependence.

Most recoverable signal is already captured by the mean. The residual can contain two components. The first is deterministic encoder entanglement: the encoder may mix otherwise independent axes in a way that a richer label-conditioned prior could, in principle, recover. The second is per-sample variation: two reviews with the same attribute combination can encode to different $\mathbf{z}$ values because of surface content, style, or wording. This component is not recoverable from labels alone and is precisely the variation removed by the conditional-mean prior.

The within-versus-between decomposition suggests that the second component is substantial. On GPT-2-Medium, the mean per-dimension standard deviation is 0.43 within a fixed combination, compared with 0.33 across per-combination means. The within-combination variance therefore sets a noise floor for any deterministic label-only predictor. A factorised prior fitted to the same checkpoint reaches residual MSE 0.20, close to this floor; Tab. 16 shows the same pattern across backbones.

Explicit interaction terms do not improve the prior. We also test whether a richer factorised prior helps in practice: a per-axis additive base plus a rank-24 interaction term. It matches the plain prior within noise on GPT-2-Medium (seen 76.4 vs. 78.5%, unseen 57.7 vs. 59.1% on Fyelp ACD) and performs worse on LLaMA-3.2 1B (seen 83.0 vs. 86.9%, unseen 60.2 vs. 69.5%), with a larger seen-to-unseen drop. We therefore retain the plain additive prior. The result should be read conservatively: in this benchmark, the extra interaction capacity does not provide reliable gains and may overfit seen combinations.

O Training schedule, loss-weight policy, and comparability policy

Four-phase loss-weight schedule. Training follows the objective in Eq. 6. Let τ_1 denote the number of LM warm-up epochs, with default $\tau_1 = 10$ for natural-language datasets and $\tau_1 = 5$ for the synthetic task. All main runs train for 25 epochs.

Backbone	Additive decomposition (% variance explained)					Concept-code variance			
	Full	sent.	cuis.	tense	gend.	intra-std	inter-std	ratio	prior MSE
GPT-2 124M	31.7	45	23	10	5	0.42	0.29	0.68	0.20
GPT-2-Med. 355M	38.3	54	38	13	6	0.43	0.33	0.76	0.20
LLaMA-3.2 1B	47.6	67	51	20	12	0.25	0.25	1.02	0.07

Table 16: Structure of the frozen-encoder concept code $\mathbf{z}$ on the Fyelp training pool. *Additive decomposition*: percentage of variance explained by an additive per-axis model $\boldsymbol{\mu} + \sum_a \Delta_a(y_a)$, and by each concatenated per-axis block alone. *Concept-code variance*: mean per-dimension within-combination and between-combination standard deviation, their ratio, and the residual MSE of a factorised label prior. Additivity rises with backbone scale, but within-combination variation is comparable to or larger than between-combination variation, suggesting a substantial sample-specific component in the encoder code.

Phase	Range	λ_c	λ_o
1 (LM warm-up)	$\tau < \tau_1$	0	0
2 (concept ramp)	$\tau_1 \leq \tau < \tau_1 + 5$	$\to 1^\dagger$	0
3 (orthogonality ramp)	$\tau_1 + 5 \leq \tau < \tau_1 + 10$	1	$\to 0.1^\dagger$
4 (full objective)	$\tau \geq \tau_1 + 10$	1	0.1

Table 17: Four-phase loss-weight schedule for Eq. 6. †Linear ramp from 0 over 5 epochs. In phase 4, the concept and orthogonality weights are fixed at $\lambda_c = 1$ and $\lambda_o = 0.1$, respectively.

Comparability policy. To keep the protocol comparisons interpretable, we use the following policies throughout the main results.

P1: Synthetic-task setting. All synthetic-task results use the 4-axis configuration (purpose $\times$ rhetoric $\times$ structure $\times$ style), excluding the topic axis.

P2: Fixed evaluation seed. Training seeds vary only in designated multi-seed runs; the evaluation seed is fixed at 42 for sampling, prior-MLP initialisation, and diagnostic pair selection.

P3: Three z-source protocols. We report classifier inversion (CLS-INV, Eq. 1), single-sample reference encoding (REF-ENC), and the label-conditioned prior (PRIOR, Eq. 2). REF-ENC is a diagnostic, not an upper bound.

P4: One stable recipe per backbone. Each backbone uses one validation-selected stable recipe, with no within-row recipe mixing. Four main backbones use the AdaLN injector with `inject_every_n` $= 1$; Qwen-2.5 1.5B uses the additive injector with $g_{\text{init}} = -3$ and `inject_every_n` $= 2$ after AdaLN proved unstable on that backbone.

P5: Unseen split as primary. Headline results use `test_unseen`. We report `test_seen` when analysing seen–unseen gaps, matched-baseline comparisons, or classifier-default artefacts.

P6: Precision recipe fixed by backbone. GPT-2 124M and Qwen-2.5 1.5B train in fp32. Qwen-2.5 0.5B, LLaMA-3.2 1B, and LLaMA-3.2 3B use a bf16 backbone with fp32 small modules, including encoder MLPs, classifier heads, injector, and prior. App. P reports the precision ablation.

P7: Phase-4 loss weights fixed. All main runs use $\lambda_c = 1$ and $\lambda_o = 0.1$ in phase 4, as shown in Tab. 17. These constants are held fixed across backbones and seeds.

Training hyperparameters. Unless otherwise specified, all main runs use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$), learning rate 5×10^{-5} for non-LoRA parameters and 2×10^{-5} for LoRA adapters, cosine decay, gradient clipping at norm 1.0, and 25 training epochs. Batch size is 16 for backbones up to 1.5B and 4 for LLaMA-3 8B due to memory constraints. The per-axis concept dimension is $d_c = 32$, with $A = 4$ for 4-axis tasks and $A = 1$ for YelpP.

All generators use LoRA rank $r = 8$ and $\alpha = 16$, targeting attention-projection modules: `c_attn` for GPT-2 and `q_proj`, `v_proj` for Qwen-2.5 and LLaMA-3.2. The label-prior MLP g_γ has one hidden layer with 128 GELU units and is fitted post-hoc for 1000 Adam steps with learning rate 10^{-3} and batch size 64. Encoding the training pool and

fitting the prior takes under 30 seconds per checkpoint on a single GPU.

P Precision-recipe (dtype) ablation

The precision recipe in P6 uses a bf16 backbone with fp32 small modules: the concept-encoder MLPs, classifier heads, injector projections, and prior MLP. We include a dtype ablation because early full-bf16 runs showed substantially lower absolute accuracy on Qwen-0.5B and LLaMA-1B. Holding the backbone in bf16 while casting the small modules to fp32 recovered this loss. Tab. 18 reports both recipes for the backbones where both runs are available.

Interpretation. The dtype recipe shifts absolute REF-ENC and PRIOR accuracies in the same direction, but leaves the prior-over-REF-ENC gap stable. We therefore treat the dtype choice as an implementation stability issue rather than a confound for the protocol comparison. The most plausible source is bf16 rounding in the small MLPs and classifier heads that operate on frozen-backbone activations; the LoRA-adapted backbone itself appears to tolerate bf16. GPT-2 124M and Qwen-2.5 1.5B were trained in full fp32 and are not included in this ablation because no bf16 counterpart was run.

Q Singleton-combo classifier-default artefact (Fyelp `test_unseen`, CLS-INV)

This appendix documents the singleton-combo classifier-default artefact referenced in Tab. 1 footnote † and §4.

Symptom (Tab. 19). The Fyelp Hold-Out idx=-0 `test_unseen` split contains a single attribute combination, `negative_male_american_present`. Classifier inversion on LLaMA-3.2 1B produces token-salad output (PPL 1181 ± 850, $27\times$ ref-enc), yet the CompMCTG official RoBERTa-large classifier reports an inflated 4-axis mean of $64.16 \pm 14.25\%$ ($n = 5$). On the seen split (39 distinct combinations), the same checkpoints score $41.27 \pm 1.56\%$ — within 1.3 pp of the random baseline 42.5% and with $9\times$ smaller cross-seed variance.

Mechanism. On degraded text, the official classifier tends to fall back to dataset-prior predictions: `negative` for sentiment, `male` for gender, `present` for tense, and a non-`american` cuisine class. Three of these defaults match the singleton unseen target, while the cuisine default does not. The result is an artificially high 4-axis mean on a one-combination evaluation set. The same pattern appears on other backbones: Qwen-2.5 1.5B shows high gender and tense but near-chance cuisine on `test_unseen`, and GPT-2 124M shows the same gender/cuisine imbalance.

Why this is not the protocol succeeding. Two independent diagnostics confirm classifier inversion has collapsed even on `test_unseen` despite the inflated 4-axis number: (i) the per-axis breakdown above does not look like a concept-conditioned generation pattern (gender is locked at 100% with zero variance; cuisine is below random; sentiment and tense show > 25 pp variance across seeds); (ii) the corresponding PPL is 1181 ± 850, roughly $27\times$ the ref-enc's 43.08 ± 2.27, consistent with the activation-level over-shoot signature of §5.2 (rel_mod amplifying from 3.3 to 85 across the 16-layer LLaMA-1B stack).

Both diagnostics agree with the seen-split number ($41.27 \pm 1.56\%$, within 0.4 pp of chance): the protocol genuinely collapses; the `test_unseen` 4-axis mean is an artefact of the single-combination evaluation set interacting with the dataset-prior default predictions of the classifier on degraded text. We therefore report the seen-split number as the headline collapse evidence in §5 and use the `test_unseen` CLS-INV numbers only as illustrations of the artefact.

R Artefact-free ACD comparison of prior and CLS-INV

The within-split prior–CLS-INV comparison on Fyelp Hold-Out idx=-0 `test_unseen` is affected by the singleton-combination classifier-default artefact described in App. Q. In contrast, ACD idx=-0 holds out half of all attribute combinations, so its `test_unseen` split provides an artefact-free within-split comparison (Tab. 20).

This within-split comparison confirms the protocol ordering in Tab. 1: on the same checkpoints and without the singleton artefact, the prior consistently recovers more compositional control than CLS-INV.

S Concept-swap fidelity

We use a concept-swap diagnostic to test whether the learned code $\mathbf{z}$ is factored across axes or only

	bf16-everywhere	bf16 + fp32 small
Synthetic 4-axis $Acc_{4ax,unseen}$ (mean)		
LLaMA-3.2 1B (ref-enc)	0.655 ± 0.098	$\mathbf{0.791 \pm 0.021}$
LLaMA-3.2 1B (prior)	0.712 ± 0.098	$\mathbf{0.840 \pm 0.026}$
Δ prior − ref-enc	$+5.7 \pm 0.8$	$\mathbf{+4.86 \pm 0.96}$
Qwen-2.5 0.5B (ref-enc)	0.424	**0.603**
Qwen-2.5 0.5B (prior)	0.497	**0.679**
Δ prior − ref-enc	+7.32	**+7.55**
LLaMA-3.2 3B (ref-enc)	≈ 0.66	**0.857**
LLaMA-3.2 3B (prior)	≈ 0.71	**0.907**
Δ prior − ref-enc	$\approx +5.6$	**+4.93**

Table 18: Precision-recipe ablation: bf16-everywhere vs. mixed precision (bf16 backbone, fp32 small modules), four-axis synthetic, `test_unseen`. Absolute ref-enc/prior accuracies depend on dtype recipe by $\sim$+18 pp; the Δ *magnitude* between prior and ref-enc stays within ± 1.0 pp on all three backbones we measured. LLaMA-1B numbers are 5-seed means (phase1c bf16-everywhere vs. phase1d fp32-small modules); Qwen-0.5B and LLaMA-3B are single seed=42 (R5/R5d and R7/R7d ablation pairs); LLaMA-3B bf16-everywhere absolute numbers are approximate as the original bf16 run was not multi-seeded.

Axis	target	seen-split prior	cls-inv unseen
sentiment	negative	$\sim$50%	$83.79 \pm 27.62\%$
gender	male	$\sim$48%	$\mathbf{100.0 \pm 0.0}\%$
cuisine	american	$\sim$20%	$5.15 \pm 6.58\%$ ‡
tense	present	$\sim$50%	$67.70 \pm 37.37\%$
4-axis mean	—	$\sim$42% chance	$64.16 \pm 14.25\%$

Table 19: Per-axis breakdown of the singleton-combo artefact (LLaMA-1B Fyelp `test_unseen` CLS-INV, $n = 5$). ‡ cuisine is the only axis whose classifier default ($\sim 1/5$) does *not* match the singleton target `american`; the resulting cell is below random and confirms the classifier defaults are not informative on token-salad input.

Backbone	CLS-INV	PRIOR	lift
GPT-2 124M	48.47	55.91	+7.44
LLaMA-3.2 1B	50.75	69.53	+18.78
Qwen-2.5 1.5B	52.82	60.48	+7.66

Table 20: Artefact-free prior–CLS-INV comparison on CompMCTG Fyelp ACD `test_unseen`, evaluated by the official RoBERTa-large classifier suite. ACD holds out half of all attribute combinations, so the evaluation does not reduce to a singleton target. The prior improves over CLS-INV by +7.4 to +18.8 pp on matched checkpoints.

Backbone	Mean joint	Grade
Synthetic 4-axis, marginal-independent attributes		
GPT-2 (R4)	0.028	ENT
Qwen-0.5B (R5d)	0.037	ENT
Qwen-1.5B (R2)	0.004	ENT
LLaMA-1B (n=5)	**0.145 ± 0.019**	**PARTIAL**
LLaMA-3B (R7d)	**0.268**	**PARTIAL**
Fyelp natural data, seed=42		
LLaMA-1B Fyelp	0.063	ENT
Qwen-1.5B Fyelp	0.047	ENT

Table 21: Concept-swap fidelity. Bins are heuristic: HIGH > 0.40, PARTIAL 0.10–0.40, and ENT < 0.10. On the synthetic task, LLaMA shows partial axis-wise fidelity while GPT-2 and Qwen remain entangled. On Fyelp, the same pattern does not transfer: both tested natural-data checkpoints remain entangled under this diagnostic.

coherent at the full combination level. For two test samples (u, v) that differ on every axis, and for an axis a, we form

$$\mathbf{z}^a_{\text{swap}} = [\mathbf{z}_v^{(0)}, \ldots, \mathbf{z}_u^{(a)}, \ldots, \mathbf{z}_v^{(A-1)}].$$

If the code is axis-factored, generation from $\mathbf{z}^a_{\text{swap}}$ should realise the axis-a label of u and the labels of v on all other axes. We report mean joint fidelity under the loose taxonomy: HIGH > 0.40, PARTIAL 0.10–0.40, and ENT < 0.10.

The synthetic task shows a family-level pattern: LLaMA reaches the PARTIAL range, increasing from 0.145 at 1B to 0.268 at 3B, whereas GPT-2 and Qwen remain entangled. On Fyelp, however, this pattern does not transfer; both tested natural-data checkpoints remain below the entanglement threshold. Although the attribute labels are marginally independent by construction (§6.2), their textual realisations can still be entangled in the

encoder representation. We therefore treat concept-swap fidelity as a supplementary diagnostic and make no interpretability claim from it in the main text.