RANDOM CHOICE IN LANGUAGE MODELS: A LINEAR CHOICE STATE AND LOW-RANK SAMPLING CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) often make highly biased choices when asked to select an item at random. We ask whether this behavior is only an output-level failure or whether models internally represent random choice as a distinct task mode that can be read and controlled. We study this question across twelve answer domains using linear probes, sparse autoencoders (SAEs), and causal interventions on internal activations. We identify a domain-general linear representation that separates random-choice requests from deterministic ones, generalizes to unseen domains and paraphrases that do not explicitly mention randomness, extends to broader arbitrary-choice contexts, and replicates across Gemma-2, Gemma-3, Llama-3.1, Qwen2.5, and Qwen3. Causal interventions on Gemma-2-9B, with a qualitative replication on Llama-3.1-8B, reveal a functional decomposition: one direction acts as a switch into the random-choice mode, whereas a separate direction shapes the resulting answer distribution. This decomposition enables a mechanism-guided intervention that improves sampling without task-specific gradient training. We further find that a trained low-rank adaptation (LoRA) policy produces low-dimensional activation changes, motivating a low-rank representation fine-tuning (LoReFT) intervention that often reaches the trained adapter's calibration regime with substantially fewer parameters, though with lower robustness on some skewed targets. Together, these results show that random-choice behavior is supported by compact, causally relevant internal structure that can be exploited for efficient sampling control.

1 INTRODUCTION

Large language models (LLMs) often produce highly biased outputs when asked to make a random choice (Hopkins et al., 2023). A recurring example is a preference for 7 in number-choice tasks; we observe similar systematic preferences in other domains, including blue for colors and October for months. Previous work has mainly treated this behavior as a behavioral *calibration* problem. We instead ask a mechanistic question: **does the model internally represent that it is being asked to make a random choice, and if so, can this representation be identified and used to control the model's sampling?**

This question is important for both practical and scientific reasons. Many applications require models to sample from a *specified* distribution rather than simply increase the sampling temperature. Examples include simulating human behavior, generating diverse content, and playing mixed strategies in games (Park et al., 2023; Meister et al., 2025). Our control experiments take the desired target distribution as given; we do not address the separate problem of estimating a human-choice distribution from data, as studied in distributional-alignment benchmarks. Random choice also provides a clean setting for studying internal task representations. The model can be asked to choose an item *randomly* or to make a *deterministic* choice (e.g., choose the largest) while the set of possible answers remains the same. This allows us to study whether the model represents the requested mode of choice separately from the answer itself.

We first identify such a representation, test whether it generalizes beyond surface cues and plays a causal role, and then ask whether the structure we find can be used to control sampling with far less than full task-specific training. Figure 1 summarizes the resulting picture.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

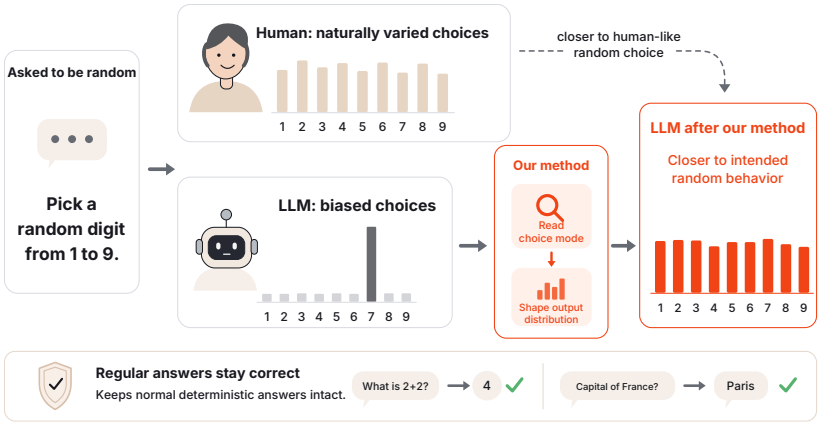

Figure 1: **Mechanism-guided control reduces biased random-choice behavior while preserving ordinary answers.** When asked to choose randomly, LLMs often concentrate probability on a small set of favored answers. Our analysis identifies an internal choice-mode signal and separates reading that state from shaping the resulting output distribution. Intervening on this mechanism produces substantially more balanced choices while leaving deterministic responses largely unchanged.

Our main findings are:

- **Choice-state representation and causal structure.** We identify a linear, domain-general choice-state representation that generalizes across domains, paraphrases, and mention controls. A sparse autoencoder (SAE) latent also reads out this state on the random-request side (§4.3; Appendix D.6). Causal interventions show that the mechanism separates into two components: the mode direction switches the random-choice regime on, while a distinct attribution-selected direction shapes the resulting answer distribution (§4.4).
- **Mechanism-guided control.** Using these two directions, we construct a hand-designed rank-1 intervention that recovers $\sim$45% of the gain achieved by the trained intervention. We evaluate this effect with gate, direction, depth, and selection controls (§5.2, Fig. 3; Appendix D.4). We find that the activation changes produced by the trained conditional LoRA adapter lie largely in a low-dimensional subspace, predicting the rank scale at which a localized representation intervention saturates. Rank-4 LoReFT reaches near pseudorandom number generator (PRNG) calibration in most model–target cells with only 187k–338k trainable parameters (§5.4).
- **Generalization and benchmarking of sampling control.** We show that the mechanism-guided low-rank control generalizes across models, target distributions, and held-out domains, while using only 187k–338k trainable parameters. We further benchmark it, together with the trained conditional LoRA ceiling, against String Seed of Thought (SSoT) under repeated finite-sample evaluation (§6.2).

2 Related Work

LLMs as random samplers. Non-uniform LLM sampling (favorite numbers, order and format effects, distributions far from target) is well documented (Hopkins et al., 2023; Meister et al., 2025; Gu et al., 2025; Koevering & Kleinberg, 2024). Most relevant is *String Seed of Thought* (SSoT) (Misaki & Akiba, 2026), a *prompting* method in which the model emits a random string and hashes it in chain-of-thought; it reaches near-PRNG faithfulness on frontier *reasoning* models but, by its authors' own analysis, is less reliable below $\sim$8B and on non-reasoning models that cannot execute the arithmetic. Our work is complementary: we give a *mechanistic* account, which SSoT lacks, and a *weight-space* controller that requires no chain-of-thought, and we use SSoT as our strongest baseline (§6.2). SSoT's own failure analysis (QwQ-32B random strings begin with "7" in $947/1000$ cases) is an independent sighting of the digit-7 bias we localize.

Fine-tuning for faithful sampling. Zhang et al. (2024) fine-tune instruction-tuned models toward diffuse output distributions, and Baldelli et al. (2026) show that probabilistic calibration is a *trainable* capability, evaluating held-out distribution families and outperforming SSoT. Distribution-matching fine-tuning and its held-out-distribution generalization are therefore prior results we build on; we do not claim the trained adapter as novel. What is new is *mechanistic*: a linearly decodable random-or-arbitrary choice state that is causally *distinct* from a downstream distribution-shaping mechanism (§4.4); a compact *mode-gated* rank-1 reader×writer edit derived from that decomposition, which the same rank-1 learner from random initialization fails to recover (§5.3); and, for representation fine-tuning (LoReFT, Wu et al., 2024), a measured activation-delta subspace that predicts the rank scale at which a localized intervention saturates (§5.4).

Concept directions, SAEs, and editing. Linear probes and difference-in-means directions recover high-level attributes from residual streams (Nanda et al., 2023; Marks & Tegmark, 2024; Zou et al., 2025), and SAEs decompose activations into interpretable latents (Huben et al., 2024; Templeton et al., 2026; Lieberum et al., 2024). We adopt the methodology of Ferrando et al. (2025), who localize whether a model *knows* an entity, but apply it to a *task mode* with an exactly measured behavioral readout; because a near-perfect probe can key on task *format* alone, we rely on leave-domain/template generalization, a token/form factorial (Appendix A.1), and a semantic factorial (Appendix D.6) rather than random splits. Activation steering (Turner et al., 2024; Rimsky et al., 2024) and function/task vectors (Todd et al., 2024; Hendel et al., 2023) use unconditional activation additions, while weight edits such as ROME (Meng et al., 2022) alter model behavior through parameter changes. Our edit instead gates the write on the mode read, and we show that for this conditional behavior an unconditional edit fires everywhere while a low-rank input-conditional adapter (Hu et al., 2022) installs the policy cleanly.

3 Experimental Setup

This section describes the experimental settings shared across Sections 4, 5, and 6.

3.1 Dataset and Prompt Design

We build 208 prompts over twelve answer domains (`digit`, `number` 1–9, `letter`, `coin`, `die`, `month`, `weekday`, `color`, `fruit`, and open-set `animal`/`city`/`word`), each pairing a *random* request ("Pick a random digit from 0 to 9") with a surface-matched *deterministic* one ("Pick the largest"). Two controls are built in: *implicit* random requests with no "random" token ("surprise me with a digit"), ruling out a token detector, and 16 *mention* prompts containing "random" but requesting no generation ("explain random access memory").

3.2 Behavioral Readout

Per domain we compute a distribution over valid answers and report its Shannon entropy H (bits) and valid-answer mass. Mechanistic analyses use a two-step whitespace/continuation marginal, whereas the probabilistic instruction following (PIF) installer benchmark in Table 2 uses the single answer-position valid-token marginal, with off-support mass reported separately (Appendix D.8). We cache the residual stream at the *last prompt token*, the position from which the answer is generated.

For probe analyses, we report ROC-AUC on continuous scores, treating random requests as the positive class. Classification accuracy uses the probe's fixed 0.5 probability threshold, with no threshold tuning.

We distinguish three evaluation protocols throughout the paper: **(P1)** the analytic or sampled valid-answer marginal at the first answer position; **(P2)** unconstrained autoregressive generation followed by parsing of the final answer; and **(P3)** full reasoning generation, where an answer is parsed only after a completed reasoning trajectory. Mechanistic and installer analyses primarily use P1; the prompting baselines and free-generation diagnostic use P2; and the open-reasoner experiment uses P3.

3.3 Models and Sparse Autoencoders

`gemma-2-9b-it`, `llama-3.1-8b-it`, and `qwen2.5-7b-it` (Gemma Team, 2024; Llama Team, 2024; Qwen et al., 2025) are the primary checkpoints used in the main analyses; `gemma-2-2b` (base) is included as an additional representation check. We additionally evaluate `gemma-3-12b-it` and `qwen3-8b` as recent-generation replications. SAE analyses use Gemma Scope JumpReLU residual SAEs (Lieberum et al., 2024), Llama Scope residual SAEs (He et al., 2024), and Gemma Scope 2 for the Gemma-3 replication.

4 Representation and Causal Structure of Random Choice

Behavioral differences alone do not establish whether models internally represent the random-choice regime. In this section, we therefore ask whether such a representation exists, whether it generalizes across domains and prompt formulations, and whether it plays a causal role in model behavior. We address these questions using linear probes, sparse autoencoders, and causal interventions.

4.1 A Domain-General Linear Representation

Behavioral premise. The original four models raise answer entropy on random requests but stay far below uniform (mean H 1.5–2.7 bits vs. the mixed-domain uniform mean of 3.09; the trained adapter reaches 3.10 because its held-out domains are larger on average), with strong favorite answers: digit 7 in three of four models. There is thus a real behavior for an internal signal to explain.

The mode is linear and general. A logistic probe on the answer-position residual separates random from deterministic requests with in-distribution AUC $= 1.0$ (bootstrap 95% CI $[1.0, 1.0]$) in all four models. This in-distribution number is *saturated*, so we do not lean on it; the evidence that the direction is a genuine task representation is its *generalization* beyond the small hand-built set used to find it. A probe trained on all-but-one domain achieves mean held-out-domain accuracy of ≥ 0.97 (across-domain SD ≤ 0.08), so the direction tracks a random-or-arbitrary-choice context across topics. The best single native-basis residual coordinate (max over dimensions at the selected layer) reaches AUC 0.96–1.00; because that is a maximum over thousands of candidates, we test it against a label-permutation null: the best-of-all-dimensions AUC under 10,000 random relabelings ranges from 0.72 to 0.73 across models, so the observed single-coordinate separability is far beyond selection chance (10,000-permutation test; 0/10,000 permutations reached or exceeded the observed statistic in all four models).

4.2 Generalization Beyond Surface Cues

A probe trained only on *explicit* prompts (containing "random") classifies *implicit* paraphrases at AUC $= 1.0$ from mid-stack onward, and its false-positive rate (FPR) on *mention* prompts falls to 0.0 by the middle layers. Early layers ($\leq$L4) behave like a token detector (high AUC but mention-FPR up to 0.88); the mid-stack signal is a genuine task representation (Fig. 4).

The same representation transfers to `gemma-3-12b-it` and `qwen3-8b`: leave-one-domain-out accuracy and explicit$\rightarrow$implicit transfer are 1.00 on both, with zero mention false positives from mid-stack onward (best layers 11 and 9), and both reproduce the early-layer token-detector vs. mid-stack concept dissociation (Appendix D.11; Appendix D.12).

4.3 A Selective SAE Feature

Encoding the answer-position residual through Gemma Scope SAEs and scoring each latent by its random-vs-deterministic firing-frequency gap (dropping latents firing on $> 2\%$ of 1,500 background text snippets sampled from the `NeelNanda/pile-10k` training split on Hugging Face,[1] a subset of The Pile (Gao et al., 2020), following Ferrando et al., 2025), we find highly selective choice-state SAE features at several depths. The cleanest, `L20F2001` in Gemma-2-9B, fires on 100% of random requests (incl. 100% implicit), 3% deterministic, 0% mention, 0% Pile (AUC $= 0.999$; Appendix C,

[1] https://huggingface.co/datasets/NeelNanda/pile-10k

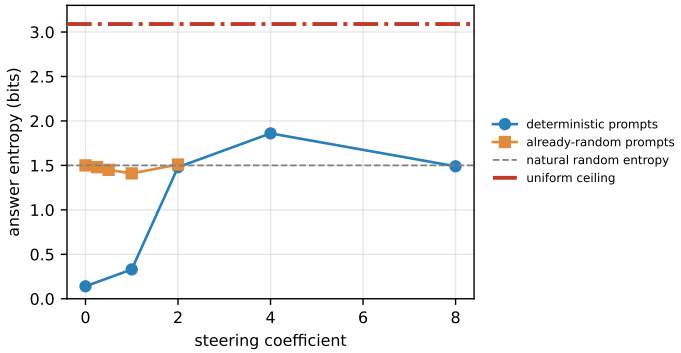

Figure 2: **The mode direction is a sufficient mode switch.** In Gemma-2-9B, the random-minus-deterministic difference-in-means direction is added at block 31. On deterministic prompts (blue), steering raises entropy to the natural random-request level; increasing the same direction on already-random prompts (orange) keeps entropy near that level, well below the uniform ceiling.

Fig. 10); the Neuronpedia public auto-interpretation (Lin, 2023) labels it a "randomness" feature, and at the answer position it tracks the *task* even when the surface token varies.

The SAE result also transfers across model and SAE generations. Using Gemma Scope 2 on `gemma-3-12b-it`, we identify `L24F5418`, which fires on 98% of random requests (100% implicit, 96% explicit), 12% of deterministic requests, 0% of mention controls, and 0.3% of Pile tokens (Appendix D.11).

4.4 Causal Structure and Cross-Model Replication

The mode direction is a sufficient switch. Figure 2 tests *sufficiency*. In Gemma-2-9B, adding the difference-in-means direction to deterministic prompts raises entropy from 0.14 to 1.48 bits, nearly matching the model's natural random-request entropy of 1.50 bits, while valid-answer mass increases. A norm-matched random direction instead reduces valid mass. Increasing the same direction on already-random prompts does not push entropy above the model's natural $\sim$1.5-bit ceiling. Thus, the direction controls entry into the random-choice regime, but does not itself make the resulting distribution uniform. The effect is also distributed across prompt positions (Appendix D.9).

A separate direction shapes the distribution. Gradient attribution, ranked by $\nabla \mathrm{H} \cdot W_{\mathrm{dec}}[j]$ among latents with positive valid-mass attribution and background firing rate below 0.05, identifies the late-layer SAE latent `L31F7399`. Ungated activation of its decoder direction raises random-request entropy from 1.50 to 2.50 bits while increasing valid mass from 0.59 to 0.76, and the entropy effect transfers to held-out domains. Under the gated mass-aware comparison of Appendix D.2, the same writer recovers 64% of the trained entropy gain at valid mass 0.66; under the stricter pre-specified reader–writer protocol of §5.2, it recovers 45%. Its natural encoder activation is not itself domain-general, so we treat it as an attribution-selected causal direction rather than a universal "randomness" feature. Together, these results separate two functions: a *reader* identifies the choice-state regime, while a downstream *writer* shapes the distribution produced within that regime.

Cross-model replication. The same qualitative decomposition appears in Llama-3.1-8B (Fig. 14; Appendix D). A selective Llama Scope feature reads the choice state (AUC 0.998, mention-FPR 0), while gradient attribution identifies a distinct entropy-increasing direction. Using the same split protocol as §5.2, the resulting hand-built reader–writer edit raises held-out random entropy from 2.03 to 2.23 bits while maintaining deterministic accuracy at 0.95, recovering 22% of the trained intervention's entropy gain. The representation transfers more consistently than the exact additive steering geometry. In `gemma-3-12b-it`, the difference-in-means direction remains linearly readable but does not produce a sufficient mode switch across seven tested depths: entropy increases only as valid-answer mass collapses, similarly to a norm-matched random control. `qwen3-8b` shows only weak, partially bidirectional steering. Thus, the random-or-arbitrary choice representation is stable across model generations, while its accessibility through a single additive direction is model-dependent (Appendix D.11; Appendix D.12).

Model	Method	H(rand, held-out)	valid mass	p_{modal}(det)
gemma-2-9b-it	ungated uniformity bias	$1.48 \rightarrow 2.11$	$0.48 \rightarrow 0.58$	$0.971 \rightarrow 0.891$
	conditional LoRA	$1.48 \rightarrow \mathbf{3.10} \pm .03$	$0.48 \rightarrow 0.90 \pm .03$	$0.971 \rightarrow 0.963 \pm .01$
llama-3.1-8b-it	conditional LoRA	$2.03 \rightarrow \mathbf{2.94} \pm .02$	$0.20 \rightarrow 0.75 \pm .05$	$0.928 \rightarrow 0.921 \pm .01$
qwen2.5-7b-it	conditional LoRA	$1.87 \rightarrow \mathbf{3.04} \pm .07$	$<0.005 \rightarrow 0.87 \pm .02$	$0.876 \rightarrow 0.800 \pm .03$
gemma-3-12b-it	conditional LoRA	$0.80 \rightarrow \mathbf{3.07} \pm .01$	$0.31 \rightarrow 0.87 \pm .04$	$0.986 \rightarrow 0.954 \pm .00$
qwen3-8b	conditional LoRA	$1.21 \rightarrow \mathbf{3.07} \pm .04$	$0.18 \rightarrow 0.80 \pm .03$	$0.905 \rightarrow 0.797 \pm .04$

Table 1: **Conditional LoRA generalizes to unseen domains.** Held-out die, month, and color domains are never used for training. Post-intervention values are means with 95% CIs across random seeds (n=5 for Gemma-2; n=3 otherwise). Conditional LoRA approaches uniform entropy while preserving deterministic behavior substantially better than an ungated uniformity-bias control.

5 Mechanism-Guided Sampling Control

The diagnosis suggests implementing the rule "when in the choice-state regime, spread over the valid answer set." We compare several ways of installing it, evaluating on three **held-out** domains (`die`, `month`, `color`) present in *neither* training objective, to separate concept-learning from answer-table memorization.

5.1 Conditional LoRA as a Trained Ceiling

Ungated uniformity bias (control). Writing the late-layer uniformity vector as a constant bias is unconditional by construction: in Gemma-2-9B it raises held-out random entropy only to 2.11 bits while dropping deterministic accuracy $0.97 \rightarrow 0.89$: a constant bias cannot gate on "is this a random request?"

Conditional LoRA. It is established that this calibration is trainable (Zhang et al., 2024; Baldelli et al., 2026); we use a trained adapter only as a behavioral ceiling for the mechanism-derived edit below, which takes no task-specific gradient steps. We train a rank-8 LoRA (Hu et al., 2022) on the late 45% of blocks with a two-part objective: on random *training-domain* prompts minimize $\mathrm{KL}(q \,\|\, \text{target}) - \lambda_{\text{mass}} \log(\text{valid mass})$ (Appendix D.8, Table 17); on deterministic and mention prompts minimize $\mathrm{KL}(p_{\text{base}} \,\|\, p_{\text{new}})$ to preserve behavior. Because the fixed LoRA update acts on input-dependent hidden states, it can learn to apply its distributional correction selectively; we test that selectivity behaviorally below (and find that the learned solution is distributed and does not substantially depend on the localized mode direction; Appendix D.1, Fig. 15). The result (Table 1; per-domain Appendix A, Fig. 9) is a near-uniform, generalizing sampler with high valid-answer mass: on *trained* domains the answer distribution reaches the uniform ceiling, and on *held-out* domains it generalizes (Gemma color $0.37 \rightarrow 3.15$, die $1.34 \rightarrow 2.57$); and deterministic modal-answer probability is largely preserved ($0.971 \rightarrow 0.963$, seed-mean).

5.2 A Hand-Built Reader–Writer Intervention

We first ask whether the discovered reader–writer structure can produce selective control without task-specific gradient training. We construct a rank-1 mode-gated edit at block ℓ=31,

$$h \mathrel{+}= \alpha \, \|\Delta\|_2 \, \hat{v}_{\text{writer}} (\hat{u}_{\text{mode}} \cdot \hat{h}),$$

where Δ is the random−deterministic difference-in-means vector estimated on the training domains, $\hat{u}_{\text{mode}} = \Delta / \|\Delta\|_2$ is its unit-normalized *reader*, $\hat{h}$ is the L_2-normalized residual, and $\hat{v}_{\text{writer}}$ is the unit-normalized attribution-selected SAE decoder direction. We retain the training-set contrast norm $\|\Delta\|_2 = 252.4$ as the intervention scale, so α is dimensionless. We use a strict train/validation/test split: the reader is estimated on training domains, α is selected on validation prompts (largest random entropy subject to validation accuracy ≥ 0.90), and results are reported once on locked test domains (die/month/color). Although the writer was initially surfaced using attribution over all domains, a train-domain-only rerun selects the same latent (Appendix D.4). With no task-specific gradient steps, the selected edit raises Gemma-2-9B held-out random entropy from 1.48 to 2.20 bits,

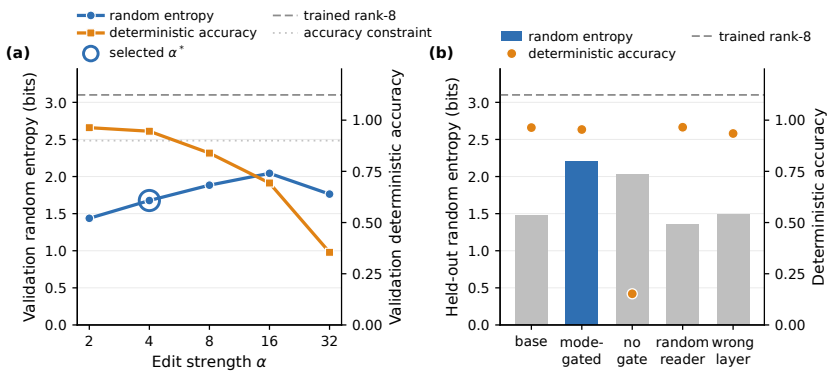

Figure 3: **The hand-built reader×writer edit yields a selective partial correction without task-specific gradient training.** The reader and edit strength α follow a strict train/validation/test split; the writer latent was independently re-selected using training domains only (Appendix D.4). **(a)** The edit strength is selected on validation domains by maximizing random-choice entropy subject to a deterministic-accuracy constraint. **(b)** On locked test domains, the selected mode-gated edit increases random-choice entropy while preserving deterministic behavior. Removing the gate destroys determinism, whereas a random reader and an early-layer edit fail to produce the same improvement. Gemma-2-9B.

recovering $\sim$45% of the trained rank-8 gain (computed on unrounded values), while preserving deterministic accuracy (0.96 $\rightarrow$ 0.95) and increasing valid-answer mass from 0.479 to 0.665 (trained ceiling 0.927). Matched controls isolate the mechanism (Fig. 3b): removing the gate collapses deterministic accuracy to 0.15, a random reader is ineffective (H = 1.36), and an early-block edit is inert (H = 1.50). The selected reader beats all 50 random-reader controls, and the selected writer beats all 50 frequency-matched-writer controls; in both cases, none of the 50 controls performed as well as the selected direction.

5.3 Mechanistic Initialization and Writer Analysis

Mechanistic initialization enables the trained rank-1 edit. We train the same rank-1 intervention ($2d$ parameters) from either the mechanistic or a random initialization. Across 10 seeds, mechanistic initialization reaches held-out entropy 2.31 ± 0.003 at valid mass 0.67, whereas random initialization reaches only 1.76 ± 0.12 at mass 0.47 (base mass 0.48). A depth sweep localizes successful training to blocks 29–31; random initialization fails there as well (Appendix D.10, Fig. 26). Thus, both the initial direction and insertion depth matter for recovering this rank-1 edit.

The trained rank-8 adapter is conditional but distributed. The adapter increases entropy by 1.62 bits on random prompts but only 0.02 on deterministic prompts, a 1.60-bit interaction (Fig. 11a). Yet projecting the localized mode direction out of the adapter branch removes only 2.1% of its entropy gain (Appendix D.1). The trained policy is therefore strongly mode-conditional behaviorally, but does not substantially depend on this localized reader direction within its own branch.

Can the writer be identified without training? We compare candidate rank-1 writers using both entropy and valid-answer mass, since entropy alone can be increased by leaking probability off the valid answer set. At block 31, the trained adapter's mean activation shift recovers 92% of its entropy gain at valid mass 0.84, exceeding the attribution-selected SAE writer (64% at mass 0.66). None of the tested base-model-only directions achieves a comparable entropy gain while passing the mass criterion (Appendix D.2, Fig. 16). Thus, our search finds a readable base-model choice-state representation but no base-only writer sufficient for full calibration.

The trained adapter's activation changes nevertheless occupy a low-rank subspace, motivating the lower-parameter representation intervention in Section 5.4.

5.4 Low-Rank Learned Control

The trained policy induces a low-dimensional residual-stream change: at Gemma-2-9B block 31, rank 8 captures $\sim$95% of its activation-delta energy, with an effective scale of roughly 8–11 dimensions (Fig. 18). This predicts saturation of a localized representation intervention at a similar rank. We test this prediction with LoReFT (Wu et al., 2024), using the same objective, data, splits, and evaluation as rank-8 LoRA (Appendix D.3). On Gemma-2-9B, rank-4 LoReFT approaches LoRA in entropy (3.03 ± 0.04 vs. 3.10), benchmark JS, preservation, and tinyGSM8K while using 272,764 rather than $\sim$7M parameters, although valid mass is lower (0.80 vs. 0.90). It also transfers to Llama, raising entropy from 2.03 to 2.93 bits.

An independent block-31 sweep is consistent with saturation around the predicted $r = 8$–11 scale, although the transition is not sharp. Single-block control plateaus near 0.70 valid mass, whereas multi-layer LoReFT composes low-rank edits across depth. Thus, our contribution is not LoReFT's known parameter efficiency, but the link between trained-policy geometry and the rank scale required for control. Under the fixed all-model PIF protocol, mean JS lies within the finite-sample PRNG reference interval in $20/25$ cells; the remaining five exhibit distribution-shape instability and/or excessive off-support mass under the fixed recipe (Table 2; Appendix D.3).

6 Generalization and Benchmarking of Sampling Control

Having established mechanism-guided sampling control in the preceding section, we next examine how broadly it generalizes and how its performance compares with prompting-based alternatives. We test the control across different target distributions, domains, models, and repeated-sampling settings, and then benchmark it against naive prompting and String Seed of Thought.

6.1 Beyond Uniform and Independent Sampling

The trained intervention is not restricted to uniform targets. On Gemma-2-9B, it matches three biased distributions with TV distance 0.018–0.021 on training domains and 0.15–0.29 on held-out domains; the result also replicates on Llama. A single prompt-conditioned adapter further generalizes jointly to unseen domains and target distributions, reducing TV from 0.51 to 0.11 on Gemma, with similar behavior on Llama and Qwen (Appendix A; Figs. 6 and 8). We also test higher-order sampling behavior following repeated-draw evaluations of distributional faithfulness (Meister et al., 2025). For repeated digit draws, the base model repeats its previous answer only 0.003 of the time, versus 0.10 under independent uniform sampling; conditional LoRA raises this to 0.093. The correlation between consecutive draws (lag-1 autocorrelation) is -0.03, consistent with weak serial dependence, and the pooled distribution reaches TV 0.024, comparable to the PRNG reference 0.026 (Appendix A; Fig. 7).

6.2 Comparison with String Seed of Thought

We compare conditional LoRA and rank-4 LoReFT with naive prompting and SSoT (Misaki & Akiba, 2026) on five PIF settings. Prompting uses N_b=20 independent K=100 free-generation batches under a pinned decoding protocol; installers use repeated K=100 draws from their valid-answer marginals, with direct target draws as the finite-sample PRNG reference. Decoding details and temperature sensitivity are in Appendices B and C.2. Conditional LoRA lies within the finite-sample PRNG reference interval in all 25 model–setting cells. It matches or improves on SSoT in all 25 cells; paired Gemma-2/Llama improvements are significant in $9/10$ cells (one tie). On the four non-trivial settings, conditional LoRA wins all 20 paired batches for both Gemma-2 and Llama (Table 8). Under the common fixed objective and mass-aware checkpoint selection, five cells—Llama biased-3, Qwen2.5 two-choice and biased-3, and Qwen3 biased-2 and biased-3—exhibit distribution-shape instability and/or excessive off-support mass under the fixed recipe and are reported without retuning. In the remaining 20 cells, mean JS lies within the finite-sample PRNG reference interval, and rank-4 LoReFT reaches the same finite-sample calibration regime as conditional LoRA while using only 187k–338k trainable parameters rather than $\sim$6–12M.

The gap to prompting is largest on harder targets. On biased-9, mean SSoT JS ranges from 31.2 to 542.1×10^{-3} across the five models, whereas conditional LoRA remains at 11.5–12.3×10^{-3}.

Model	Method	2-ch.	bias-2	3-ch.	bias-3	bias-9	params
gemma-2-9b-it	naive	11.4	117.3	150.7	117.3	300.2	—
	SSoT	4.3	40.3	25.2	20.2	31.2	—
	conditional LoRA	1.3	1.7	2.2	2.6	11.8	∼7M
	LoReFT r=4	1.3	1.3	2.5	2.0	11.1	273k
llama-3.1-8b-it	naive	2.6	311.1	211.1	344.3	475.8	—
	SSoT	2.6	58.0	53.2	188.0	396.2	—
	conditional LoRA	1.3	1.4	2.3	2.4	11.9	∼6M
	LoReFT r=4	1.3	1.5	2.5	117.3†	10.9	246k
qwen2.5-7b-it	naive	215.8	342.0	318.3	108.4	300.2	—
	SSoT	99.0	124.6	210.6	304.0	249.5	—
	conditional LoRA	1.2	1.5	2.1	2.5	11.8	∼7M
	LoReFT r=4	57.0†	1.4	2.5	71.6†	11.1	187k
gemma-3-12b-it	naive	215.8	117.3	67.7	117.3	300.2	—
	SSoT	60.9	76.3	160.2	83.8	59.1	—
	conditional LoRA	1.3	1.9	2.2	2.5	12.3	∼12M
	LoReFT r=4	1.3	1.8	2.5	2.0	10.9	338k
qwen3-8b	naive	215.8	241.1	318.3	307.7	348.2	—
	SSoT	69.3	274.8	242.7	360.1	542.1	—
	conditional LoRA	1.3	1.6	2.5	2.5	11.5	∼8M
	LoReFT r=4	1.3	1.3†	2.5	32.8†	11.0	279k
PRNG finite-sample reference		1.0	1.5	2.9	2.1	9.6	—

Table 2: **Probabilistic Instruction Following.** Mean JS divergence to target ($\times 10^{-3}$; lower is better) over repeated K=100 samples. Prompting uses N_b=20 full-generation batches; LoRA and LoReFT are sampled from installed valid-answer marginals. † marks LoReFT cells exhibiting distribution-shape instability and/or excessive off-support mass under the fixed recipe; full intervals, parse rates, off-support mass, and seed-level results are in Appendices D.8 and D.3.

The installed conditional-LoRA distributions themselves have analytic JS 0.00–0.24 $\times$ 10^{-3}, so most of the remaining empirical divergence at K=100 reflects finite-sample variability rather than calibration error. Across the original Gemma-2, Llama-3.1, and Qwen2.5 benchmark, naive prompting produces 13.4% invalid outputs and SSoT 10.3%. The distinction between restricted-marginal and end-to-end evaluation is substantive but does not eliminate the behavioral effect. Under unconstrained free generation, the adapted model improves over a target-in-prompt naive base in 9/10 Gemma-2/Llama cells (Appendix D.8). On `DeepSeek-R1-Distill-Llama-8B`, it likewise improves over the matched same-prompt base model in all four successfully trained settings, although calibration after a full stochastic reasoning trajectory is substantially weaker than at the analytic answer-position readout (Appendix B.2). The valid-mass objective reduces off-support probability from 0.35 to 0.0004 while keeping conditional-on-valid TV below 0.02 (Fig. 12; Appendix D.8).

7 Conclusion and Limitations

Models encode a domain-general, linearly decodable *random-or-arbitrary-choice state* that generalizes beyond the trigger token across Gemma-2/3, Llama-3.1, and Qwen2.5/3. SAE features recover its random-request side, while the residual-stream direction captures broader semantics. Controlling behavior requires a separate distribution-shaping writer; the representation is more stable across models than any particular additive steering geometry. Across five models spanning three model families, rank-4 LoReFT reaches the finite-sample PRNG regime in 20/25 cells with 187k–338k parameters; the other five show instability or excessive off-support mass. Limitations include nine small closed-set domains, primarily first-position evaluation, and no continuous targets. The mode direction does not distinguish stochastic from arbitrary choice (AUC 0.50/0.57 on Gemma/Llama; Appendix D.6). The rank-1 writer mainly increases diffuseness; tested SAE-free writers do not recover the effect (Appendix D.7). Capability retention is assessed only with deterministic accuracy and tinyMMLU/tinyGSM8K. End-to-end generation, including stochastic reasoning, is less calibrated; Qwen models show a task-local preservation cost.

Reproducibility Statement

All models and SAEs are public checkpoints (§3; Table 5). The 208-prompt design, the implicit and mention controls, and the factorial and naturalistic control sets are described in §3 and Appendices A.1–A.2. The PIF settings, prompting protocols, semantic controls, target definitions, and parsing rules are documented in the corresponding appendices. Training configurations for conditional LoRA, LoReFT, and the trained rank-1 edit (objective, adapted blocks, learning rates, mass-reward weights, step counts, seeds, and domain splits) are given in Appendix B and Table 17; the hand-built edit's split discipline and selection controls are in §5.2 and Appendix D.4; the LoReFT intervention and rank sweeps are in Appendix D.3; and the benchmark protocol, PRNG reference, temperature sweep, and paired-batch comparison are in Appendices D.8 and C.2.

AI Use Statement

Generative AI tools were used for the following tasks with required disclosure: generating synthetic data, specifically the 404 paraphrastic variants in the factorial-control set (Appendix A.1), produced by a general-purpose assistant model under a meaning-preserving instruction and subsequently filtered programmatically (model version and decoding settings were not logged, so this step is not exactly reproducible; the resulting prompt set is fixed and used unchanged in all experiments).

Generative AI tools were used for feedback on methodology, experimental controls, follow-up diagnostics, presentation, interpretation, coding assistance, and manuscript editing. Final experimental decisions, implementations, analyses, scientific interpretations, and claims were determined and verified by the authors.

The models studied here (Gemma-2-9B, Gemma-2-2B, Llama-3.1-8B, Qwen2.5-7B, Gemma-3-12B, Qwen3-8B, and DeepSeek-R1-Distill-Llama-8B) were not used to construct or label the evaluation data.

We have reviewed all AI-assisted work: AI-assisted code was checked by the authors, and all reported numbers, tables, and figures were produced by our analysis code and verified against saved experimental artifacts. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

Davide Baldelli, Sruthi Kuriakose, Maryam Hashemzadeh, Amal Zouaq, and Sarath Chandar. Probabilistic calibration is a trainable capability in language models, 2026. URL https://arxiv.org/abs/2605.11845.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 82318–82355. Curran Associates, Inc., 2025. doi: 10.52202/085713-2755. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/764ff7477b8e24dbe01531f6791e8bdf-Paper-Conference.pdf.

Javier Ferrando, Oscar Balcells Obeso, Senthooran Rajamanoharan, and Neel Nanda. Do i know this entity? knowledge awareness and hallucinations in language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=WCRQFlji2q.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020. URL https://arxiv.org/abs/2101.00027.

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Jia Gu, Liang Pang, Huawei Shen, and Xueqi Cheng. Do LLMs play dice? exploring probability distribution sampling in large language models for behavioral simulation. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5375–5390, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.360/.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders, 2024. URL https://arxiv.org/abs/2410.20526.

Thomas Heap, Tim Lawson, Lucy Farnik, and Laurence Aitchison. Automated interpretability metrics do not distinguish trained and random transformers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=USyGD0eUod.

Lovis Heindrich, Philip Torr, Fazl Barez, and Veronika Thost. Do sparse autoencoders generalize? a case study of answerability. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2025. URL https://openreview.net/forum?id=rs3alQ5BV8.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.624. URL https://aclanthology.org/2023.findings-emnlp.624/.

Aspen K Hopkins, Alex Renda, and Michael Carbin. Can LLMs generate random numbers? evaluating LLM sampling in controlled domains. In *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*, 2023. URL https://openreview.net/forum?id=Vhh1K9LjVI.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Robert Huben, Hoagy Cunningham, Logan Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 7827–7845, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1fa1ab11f4bd5f94b2ec20e794dbfa3b-Paper-Conference.pdf.

Katherine Van Koevering and Jon Kleinberg. How random is random? evaluating the randomness and humaness of llms' coin flips, 2024. URL https://arxiv.org/abs/2406.00092.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 278–300, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.19. URL https://aclanthology.org/2024.blackboxnlp-1.19/.

Johnny Lin. Neuronpedia: Interactive reference and tooling for analyzing neural networks, 2023. URL https://www.neuronpedia.org. Software available from neuronpedia.org.

Llama Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets, 2024. URL https://arxiv.org/abs/2310.06824.

Nicole Meister, Carlos Guestrin, and Tatsunori Hashimoto. Benchmarking distributional alignment of large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 24–49, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.2. URL https://aclanthology.org/2025.naacl-long.2/.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022. doi: 10.52202/068431-1262. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.

Kou Misaki and Takuya Akiba. String seed of thought: Prompting LLMs for distribution-faithful and diverse generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=luXtbX1lVK.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi (eds.), *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.2. URL https://aclanthology.org/2023.blackboxnlp-1.2/.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet, 2026. URL https://arxiv.org/abs/2605.29358.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron Wallace, and David Bau. Function vectors in large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 17282–17333, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4ae163cb8788970e53b4fd9578141139-Paper-Conference.pdf.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. URL https://arxiv.org/abs/2308.10248.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 63908–63962. Curran Associates, Inc., 2024. doi: 10.52202/079017-2041. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/75008a0fba53bf13b0bb3b7bff986e0e-Paper-Conference.pdf.

Yiming Zhang, Avi Schwarzschild, Nicholas Carlini, J Zico Kolter, and Daphne Ippolito. Forcing diffuse distributions out of language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=9JY1QLVFPZ.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2025. URL https://arxiv.org/abs/2310.01405.

A ADDITIONAL BENCHMARK AND PROBE RESULTS

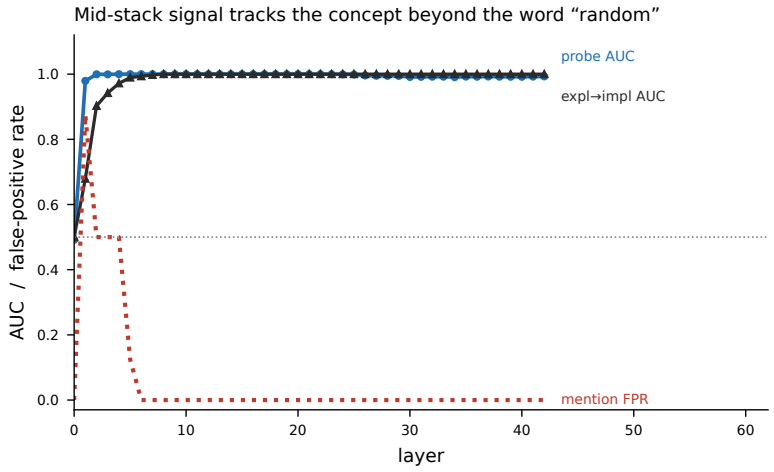

Figure 4: Concept-not-token dissociation in Gemma-2-9B. Early layers achieve high probe AUC but high mention false-positive rate (a “random”-token detector); by mid-stack the probe transfers perfectly to implicit paraphrases while its mention-FPR drops to zero.

A.1 FACTORIAL TOKEN/FORM CONTROLS

We test whether the probe is detecting the intended choice mode rather than simply recognizing the word “random.” We construct a 971-prompt factorial dataset in which choice intent is varied independently of the literal “random” token, prompt form (imperative, question, or dialogue), and answer domain. Crucially, the dataset includes prompts that break the usual correlation between wording and intent: for example, “pick any digit you like” expresses arbitrary choice without using the word “random,” whereas “do not choose randomly; give the largest” contains the word but requests a deterministic answer. The prompts are generated programmatically from 21 author-written templates; we additionally include 404 paraphrastic variants generated by an assistant model and validated by filtering (see the AI Use Statement).

Table 3 reports linear-probe AUC for predicting the random-or-arbitrary choice state. Presence of the literal “random” token alone is uninformative (AUC 0.46). In contrast, the mid-stack probe (best layer 8) generalizes almost perfectly when either the answer domain (AUC 0.9998) or prompt template (AUC 0.999) is held out. It also reaches AUC 1.00 when evaluated separately on prompts that contain “random” and on prompts that do not, showing that its prediction does not depend on that single token.

This control does not establish that the representation is entirely independent of lexical cues. On these short, highly structured prompts, a bag-of-words classifier also reaches AUC 0.999, indicating that other words in the prompt remain strongly predictive of intent. We therefore complement this factorial test with the more naturalistic evaluation below.

A.2 NATURALISTIC PROMPTS REDUCE LEXICAL CONFOUNDS

We next test whether the mode representation persists when simple lexical cues are made much less informative. We hand-author 10 scenario specifications and expand them deterministically into 180 naturalistic, multi-sentence prompts (10 scenarios $\times$ 9 domains $\times$ 2 intents). Within each scenario, the random and deterministic versions share the same context. Their closing instructions are also deliberately written to reuse vocabulary associated with the opposite class: random requests mention the deterministic rule, and deterministic requests reuse language associated with random choice. This substantially increases lexical overlap between the two classes (vocabulary Jaccard 0.77).

We compare the answer-position linear probe against a bag-of-words (BOW) classifier (Fig. 5). In a *leave-scenario-out* evaluation, where all wording from one scenario is held out, the probe reaches AUC 1.00 on both models, whereas BOW falls to 0.90. BOW nevertheless reaches AUC 1.00 under ordinary in-distribution cross-validation, suggesting that its performance depends more strongly on scenario-specific wording.

We then perform a harder *cross-distribution transfer* test. Here, we freeze the mode direction learned from the original short-template prompts and apply it directly to the naturalistic prompts. Llama transfers at AUC 0.93, compared with 0.86 for BOW. Gemma reaches AUC 0.82 at the preselected frozen layer, although its median across layers is 0.95. Thus, the leave-scenario-out result provides strong evidence that the probe is not merely exploiting scenario-specific lexical cues, while the cross-distribution result is supportive but less uniform across models. Together, these tests address the lexical confound that remains in the short-template factorial experiment.

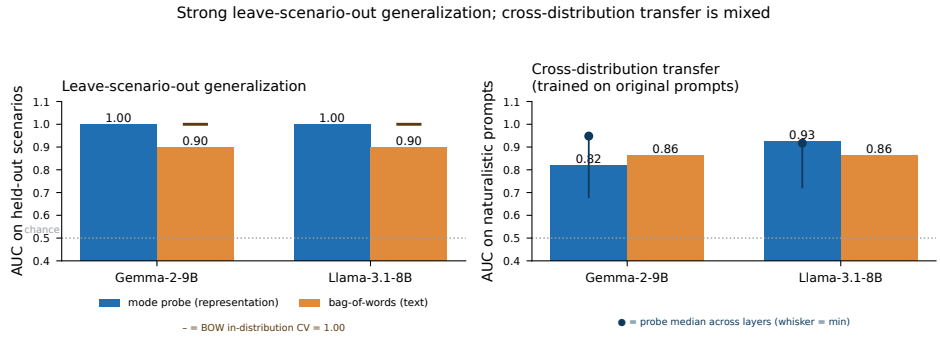

Figure 5: **Strong leave-scenario-out generalization; cross-distribution transfer is mixed.** **(a)** Leave-scenario-out: the mode probe holds at 1.00 while BOW falls to 0.90 (dash = BOW in-distribution CV = 1.00). **(b)** Cross-distribution transfer from the original prompts (dot/whisker = probe median/ min across layers).

predictor / split	AUC
activation, random 5-fold CV	1.000
activation, leave-domain-out	0.9998
activation, leave-template-out	0.999
activation, within "random" absent (n=640)	1.000
activation, within "random" present (n=331)	1.000
bag-of-words baseline	0.999
prompt-length baseline	0.732
"random" token only	0.458

Table 3: Factorial token/form controls (Gemma-2-9B, $n = 971$). AUC for predicting random-or-arbitrary-choice intent.

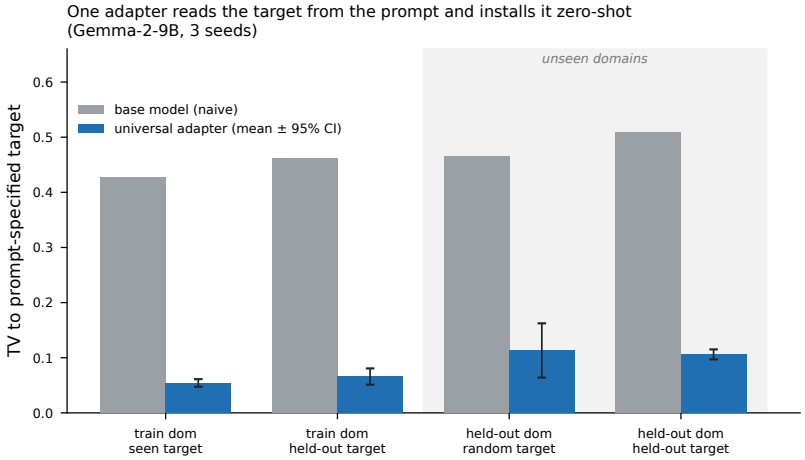

Figure 6: **One prompt-conditioned adapter generalizes to unseen targets and domains.** The adapter reads the requested distribution from the prompt and installs it without retraining (Gemma-2-9B, mean ± 95% CI over 3 seeds). Grey: base model; blue: adapter. The right two conditions use held-out answer domains.

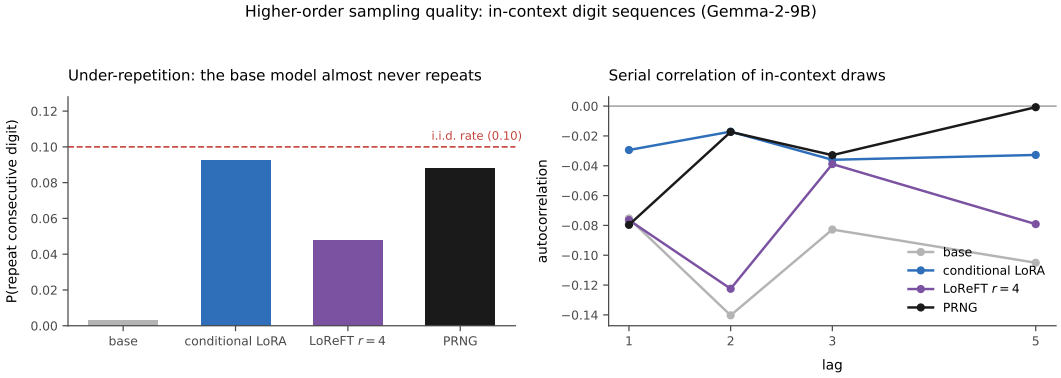

Figure 7: **Higher-order quality of in-context digit sequences (Gemma-2-9B).** Left: the base model almost never repeats the same digit on consecutive draws. Conditional LoRA nearly restores the repetition rate expected from independent uniform sampling, while LoReFT provides a partial correction. Right: Conditional LoRA also removes most serial dependence: correlations between draws separated by one or more positions stay close to zero, as expected for independent samples. LoReFT again improves over the base model but remains less consistent. PRNG is shown as a finite-sample reference.

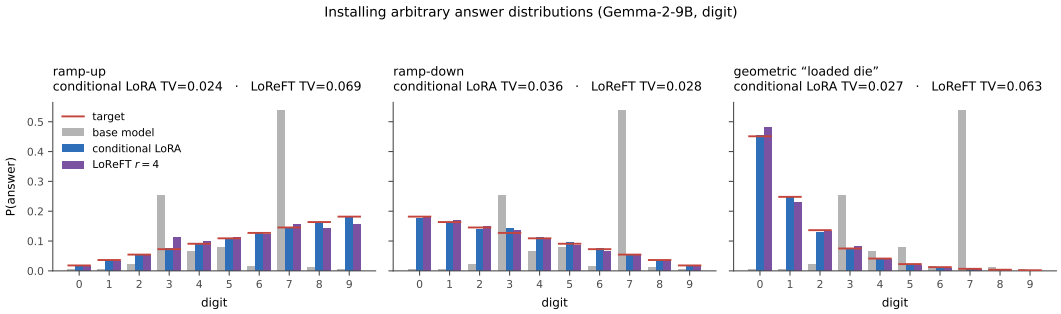

Figure 8: **Installing arbitrary answer distributions (Gemma-2-9B, digit).** Conditional LoRA and rank-4 LoReFT are trained to match three non-uniform target distributions: an increasing ramp, a decreasing ramp, and a geometric "loaded-die" distribution. Red lines mark the target probabilities, and per-panel TV distances are reported above each plot. The geometric target is deliberately far from uniform, showing that the interventions can install structured bias rather than merely flattening the output distribution.

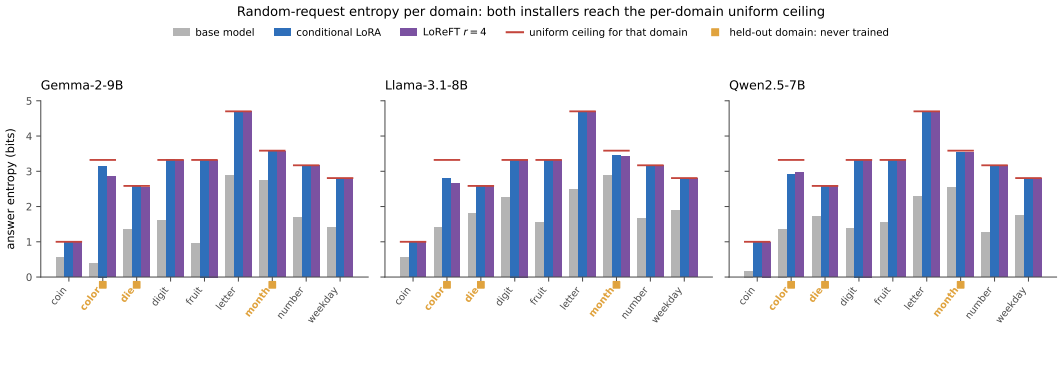

Figure 9: **Per-domain random-request entropy.** Conditional LoRA reaches the uniform ceiling across domains, while rank-4 LoReFT closely approaches it. Red ticks mark the uniform ceiling; orange squares mark held-out domains.

A.3 General-capability retention (MMLU)

The main text verifies that the installers largely preserve deterministic-task behavior. Because the interventions modify late residual blocks across all positions, we additionally test whether they affect general knowledge outside the answer domains. We evaluate the base model, conditional LoRA, and rank-4 LoReFT on tinyMMLU (100 five-shot questions, scored by the argmax next-token log-probability over option letters A–D) across all five evaluated models.

Table 4: **tinyMMLU retention.** Accuracy on 100 five-shot questions, scored by letter argmax. Conditional LoRA preserves performance across the evaluated models, while LoReFT shows only small changes.

Model	base	cond. LoRA	LoReFT r=4
gemma-2-9b-it	0.76	0.75	0.77
llama-3.1-8b-it	0.62	0.62	0.63
qwen2.5-7b-it	0.77	0.77	0.77
gemma-3-12b-it	0.67	0.67	0.67
qwen3-8b	0.77	0.77	0.75

Conditional LoRA preserves tinyMMLU across the evaluated checkpoints, and LoReFT shows only small changes on this probe (Table 4). In particular, Qwen2.5's deterministic-task accuracy decreases from 0.88 to 0.80, while its tinyMMLU accuracy remains unchanged at 0.77, suggesting that this cost is task-local rather than a broad loss of knowledge. Because tinyMMLU contains only 100 items, changes of approximately 0.01 should not be interpreted as meaningful improvements (Table 4).

B LoRA training defaults and evaluation settings

Table 5 gives the conditional-LoRA configuration and the pinned decoding protocol used for the repeated PIF baseline evaluation. LoReFT and the trained rank-1 intervention are documented separately in Appendices D.3 and D.10. Models are loaded in `bfloat16`; no base-model weights are updated.

Component	Value
Base model (mechanism)	`google/gemma-2-9b-it` (bf16)
Base models (replication)	`meta-llama/Llama-3.1-8B-Instruct`, `Qwen/Qwen2.5-7B-Instruct`, `google/gemma-3-12b-it`, `Qwen/Qwen3-8B`
Adapter	LoRA, rank r=8, α=16, dropout 0.0
Adapted modules	`gate_proj, up_proj, down_proj, o_proj`
Adapted blocks	late 45%: `layers_to_transform`= $[0.55n], \ldots, n$ (23–41 for Gemma-2-9B)
Optimizer	AdamW, lr 2×10^{-4}, 300 steps (250 for original PIF adapters)
Batch composition	6 random-domain + 6 preservation prompts per step
Objective	$\mathbb{E}_{\text{rand,train}}[\mathrm{KL}(q\|\|\text{target}) - \lambda_{\text{mass}} \log(\text{valid mass})]$ $+\lambda\, \mathbb{E}_{\text{det+mention}} \mathrm{KL}(p_{\text{base}}\|\|p_{\text{new}})$
Loss coefficients	default λ_{mass}=0.05, λ=1.0; experiment-specific values reported with each experiment
Domain split (rank-1)	train {digit, number, letter, coin}; val {weekday, fruit}; test {die, month, color}
Domain split (LoRA)	held-out {die, month, color}; never used in the training objective
Mechanistic readout	two-step whitespace/continuation marginal at the last prompt token
Prompting sampler	`do_sample=True`, `top_p=0.95`, `top_k=0`, repetition penalty 1.0, `num_beams=1`
Prompting temperature	Gemma-2 1.0; Llama-3.1 0.6; Qwen2.5 0.7; Gemma-3 1.0; Qwen3 0.7
Generation length	`max_new_tokens=24` (naive); `320` (SSoT)
Prompting evaluation	N_b=20 independent batches of K=100 full generations
Qwen3 prompting mode	`enable_thinking=False`
PIF installer evaluation	single answer-position valid-token marginal at native T=1; no top-k/top-p truncation; off-support mass reported separately
Loading	`bfloat16`, `attn_implementation=eager`

Table 5: **Training and evaluation configuration.** The prompting rows give the pinned protocol used for the repeated PIF baseline reruns in Table 2; Appendix C.2 reports a separate temperature sensitivity analysis.

B.1 Exact Benchmark Prompts and Parsing Rules

For reproducibility, we report the exact prompt templates used in the PIF benchmark. The target distribution is exposed to the prompting baselines but not to the trained installers. Decoding parameters are reported in Appendix B.

PIF settings and targets. Probability vectors follow the answer ordering shown below.

Trained-installer evaluation prompts. Conditional LoRA and LoReFT receive only the bare random-choice prompt. The desired target distribution is supplied by the learned intervention and does not appear in the prompt.

Table 6: **PIF benchmark settings and target distributions.** Probability vectors follow the answer ordering shown in the middle column.

Setting	Answers (in order)	Target distribution
2-choice	heads, tails	$[0.5,\ 0.5]$
biased-2	heads, tails	$[0.3,\ 0.7]$
3-choice	rock, paper, scissors	$[1/3,\ 1/3,\ 1/3]$
biased-3	rock, paper, scissors	$[0.1,\ 0.2,\ 0.7]$
biased-9	one,. . . ,nine	$[0.08^{\times 8},\ 0.36]$

2-choice / biased-2

```
Flip a fair coin and tell me the result:  heads or tails?
```

3-choice / biased-3

```
Throw rock, paper, or scissors at random.
```

biased-9

```
Pick a random number word from one to nine.
```

Training uses four random-choice phrasings for each answer family; the prompts above are the canonical evaluation phrasings.

Naive prompting. Naive prompting uses the following system instruction.

System message

```
Select exactly one option from the given list.  End with the final
choice on its own line as:  ANSWER: <option>.
```

For uniform targets, the user instruction is:

Uniform-target user message

```
Choose uniformly at random among:  {options}.
```

For non-uniform targets, the user instruction is:

Biased-target user message

```
Choose one of:  {options}, so your long-run frequencies match these
target probabilities -> {option_1}:  {prob_1}, {option_2}:  {prob_2},
...
```

For example, the biased three-choice setting is instantiated as:

Biased-3 instantiated example

```
Choose one of:  rock, paper, scissors, so your long-run frequencies
match these target probabilities -> rock:  0.10, paper:  0.20,
scissors:  0.70.
```

String Seed of Thought. SSoT uses the same user instruction as naive prompting and the following system instruction.

SSoT system message

```
You must select exactly one option from a given list, so that over
many independent calls your choices follow the target probabilities.
To avoid your habitual biases, first generate a fresh random
string of about 24 characters (letters, digits, symbols) that
looks arbitrary.  Then convert it to a number by summing the ASCII
codes of all characters.  Use that number, together with the target
probabilities, to pick the option (e.g.  take the sum modulo 100 and
compare against the cumulative probabilities).  Show the string and
the arithmetic, then end with the final choice on its own line as:
ANSWER: <option>.
```

There are no model-specific SSoT prompt variants.

Chat-template wrapping. Prompting baselines use each model's native chat template with a generation prompt. For Gemma-2 and Gemma-3, which do not accept the system role used here, the system and user strings are concatenated with a blank line into a single user turn. Qwen3 is evaluated with thinking disabled.

Target-in-prompt and matched-base controls. The target-in-prompt naive baseline uses the full naive prompt above, including the target distribution. This differs from the matched-base condition in the reasoning experiment: the matched base receives exactly the same bare random-choice prompt as the adapted model and receives no target probabilities.

Reasoning-model prompt. For `DeepSeek-R1-Distill-Llama-8B`, the adapted model and matched base receive the same bare random-choice prompt as above, with no system message and no additional instruction to reason, emit `<think>`, or use a particular answer format. The model therefore follows its native reasoning behavior.

Valid answers and analytic scoring. Valid answer strings follow the answer ordering in Table 6. The analytic valid-token readout aggregates case and leading-space variants: lowercase, capitalized, and uppercase forms, each with and without a leading space. Token ids shared by multiple answers are excluded from both answers; no answer in the audited PIF settings has an empty token set.

Free-generation parsing. For naive prompting, SSoT, and ordinary adapted free generation, parsing first examines text following a case-insensitive `ANSWER:` marker, or the whole completion when that marker is absent. Whole-word answer matches are case-insensitive. If this fails, the parser falls back to the last occurrence of any valid answer string in the completion. Unparseable generations are excluded from the empirical answer distribution, and parse rate is reported separately.

For the reasoning-model experiment, parsing is stricter. A completion is eligible only if it contains a closed `</think>` block, and only text after the first closing tag is searched for an answer. Generations without a closing tag are counted as truncated rather than parsed from the reasoning trace.

B.2 End-to-End Sampling Through a Reasoning Trajectory

The restricted answer-position evaluation used in the main PIF benchmark does not capture randomness introduced by a model's subsequent reasoning trajectory. We therefore run a separate end-to-end experiment on `DeepSeek-R1-Distill-Llama-8B`. The corrected protocol allows up to 2048 generated tokens, requires a closed `</think>` block, and parses an answer only from the post-reasoning text. Truncated chains and generations with no post-reasoning answer are recorded rather than parsed from intermediate chain-of-thought. We use $K{=}64$ generations per condition.

For each setting we compare three quantities. The *analytic* readout is the installed valid-answer distribution at the answer-position representation; the *adapter free+CoT* condition allows the adapted model to reason and generate normally; and the *matched base* runs the unadapted model on the same bare prompt. The reasoning-model installer uses the mass-corrected objective with $\lambda_{\text{mass}}{=}0.5$,

mass-aware checkpoint selection, and 1000 training steps. Table 7 reports the resulting end-to-end comparison.

Setting	trained	analytic TV	adapter + CoT TV	matched-base TV	parsed / 64
2-choice uniform	yes	0.000	0.329	0.415	35
2-choice biased	yes	0.005	0.537	0.615	49
3-choice uniform	yes	0.003	0.176	0.633	55
3-choice biased	no	0.900	0.864	0.866	55
9-choice biased	yes	0.003	0.510	0.703	44

Table 7: **End-to-end sampling through stochastic reasoning on DeepSeek-R1-Distill-Llama-8B.** TV is measured against the requested target distribution. "Analytic" is the installed answer-position distribution before a reasoning trajectory; "adapter + CoT" and "matched base" are full stochastic generations from the adapted and unadapted models on the same prompt. Free-generation TV is computed over strictly parsed post-`</think>` answers, so the final column reports parseability explicitly. The adapted model improves over the matched base in all four successfully trained settings, but the substantial gap between analytic and end-to-end TV shows that stochastic reasoning introduces additional distributional error. The biased three-choice installer did not train successfully and is not counted as evidence for end-to-end control.

Among the four successfully trained settings, the adapted model also improves over the target-in-prompt naive base in three; biased-2 is the exception, where the unadapted model's native full-generation distribution happens to lie closer to the requested target. Thus, control at the answer-position marginal does survive a reasoning trajectory, but does not determine the final empirical distribution once stochastic chain-of-thought generation is introduced.

C Additional Analyses and Figures

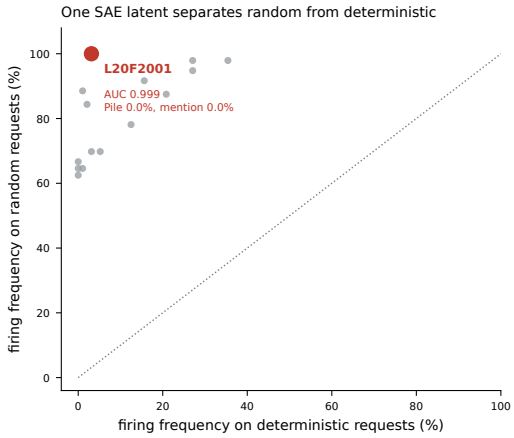

Figure 10: Top choice-state SAE latents in Gemma-2-9B (block 20). `L20F2001` fires on essentially every random request and almost no deterministic one (AUC $= 0.999$), and is silent on "random"-mention and Pile inputs.

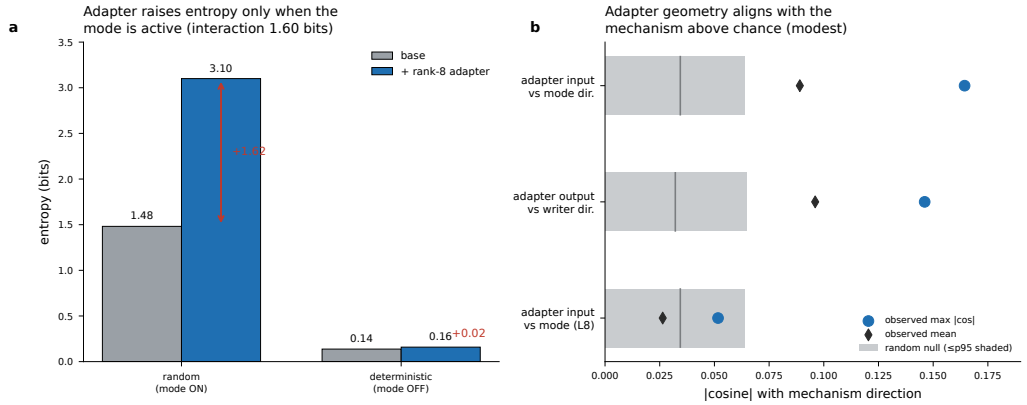

Figure 11: **The trained rank-8 adapter is behaviorally conditional but uses a distributed solution.** **(a)** It raises entropy by 1.62 bits on random prompts but only 0.02 bits on deterministic ones (interaction 1.60 bits), showing strongly prompt-conditional behavior. **(b)** Its input and output directions overlap with the localized reader and attribution-selected writer only modestly above a random-direction null (grey $\leq p_{95}$; tick = max). Together with the branch-only ablation (Fig. 15), this indicates that the trained adapter does not depend on the localized reader direction identified by the hand-built intervention, but reaches its conditional behavior through a distributed solution.

We report both the end-to-end invalid-outcome analysis (Fig. 12) and the corresponding exact-probability analysis (Fig. 13).

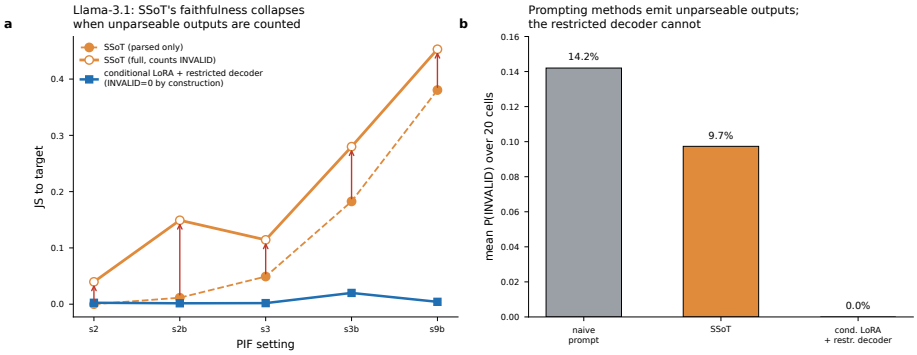

Figure 12: **End-to-end INVALID outcome (parse rate).** **(a)** On Llama-3.1 SSoT's JS to target rises sharply once unparseable outputs are counted (dashed = parsed only; solid = full). **(b)** Mean $P(\text{INVALID})$ over 20 cells (four models $\times$ five settings, including the reasoning model that is excluded from the headline 15-cell benchmark) for the free-generation methods. The weight-level comparison is Figure 13.

C.1 PAIRED MULTI-BATCH COMPARISON WITH SSOT

To make the operating-point comparison in Table 2 statistically explicit, we evaluate SSoT and conditional LoRA with N_b=20 independent batches of K=100 samples. SSoT uses the pinned full-generation protocol, whereas conditional LoRA is sampled from the installed valid-answer marginal. Table 8 reports the mean JS divergence, the central 95% interval across the 20 batches, and the fraction of paired batches in which conditional LoRA obtains lower JS. We additionally bootstrap the paired batchwise difference $\text{JS}_{\text{LoRA}} - \text{JS}_{\text{SSoT}}$.

On all four non-trivial settings, conditional LoRA wins all $20/20$ paired batches for both Gemma-2 and Llama. On the two-choice uniform setting, Gemma-2 favors conditional LoRA in $17/20$ batches, while Llama favors it in $14/20$; the paired difference on Llama includes zero and is therefore treated as a statistical tie.

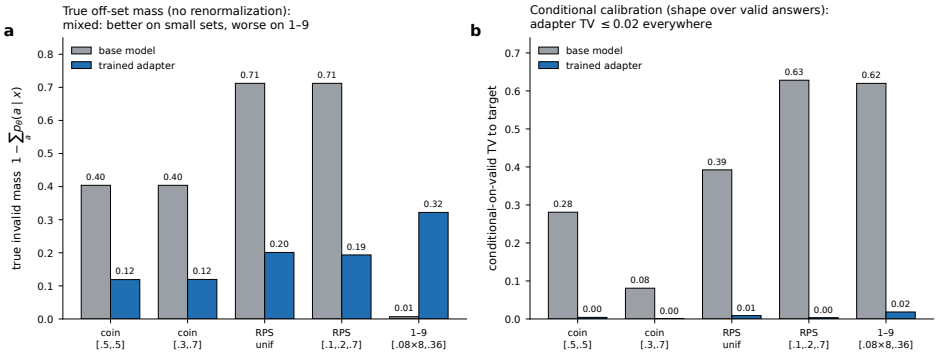

Figure 13: **Exact-probability track (no restricted sampler, no renormalization). (a)** True invalid mass $1 - \sum_a p_\theta(a \mid x)$ from the model's own distribution: the adapter reduces off-support mass on the coin/RPS sets but, under the original objective, increases it on the nine-way task. The mass-corrected objective used throughout the paper removes this regression (off-support mass ≤ 0.004; Table 2). **(b)** Conditional-on-valid TV to target: the adapter is near-perfectly calibrated over the valid answers everywhere ($0.40 \rightarrow 0.007$ mean). Gemma-2-9B.

Model	Method	2-choice	biased-2	3-choice	biased-3	biased-9
gemma-2-9b-it	SSoT	4.3 [0.4,12.4]	40.3 [23.4,63.0]	25.2 [8.9,42.3]	20.2 [8.9,41.5]	31.2 [14.2,61.6]
	conditional LoRA	1.3 [0.1,6.2]	1.7 [0.0,6.7]	2.2 [0.2,5.6]	2.6 [0.2,7.6]	11.8 [3.8,29.2]
	Pr(LoRA win)	0.85	1.00	1.00	1.00	1.00
	paired Δ CI	[-4.72,-1.34]	[-43.94,-33.85]	[-27.42,-18.69]	[-22.09,-13.56]	[-26.32,-13.05]
llama-3.1-8b-it	SSoT	2.6 [0.0,7.1]	58.0 [34.8,91.0]	53.2 [28.5,83.0]	188.0 [161.8,224.6]	396.2 [319.3,465.5]
	conditional LoRA	1.3 [0.1,6.2]	1.4 [0.0,5.7]	2.3 [0.2,6.2]	2.4 [0.2,6.4]	11.9 [4.0,29.4]
	Pr(LoRA win)	0.70	1.00	1.00	1.00	1.00
	paired Δ CI	[-2.65,0.08]	[-64.06,-49.87]	[-57.51,-44.27]	[-193.72,-178.01]	[-403.34,-365.13]

Table 8: **Paired multi-batch comparison of SSoT and conditional LoRA.** JS divergence is reported in units of $\times 10^{-3}$ as the mean followed by the central 95% interval across N_b=20 independent K=100 batches. Pr(LoRA win) is the fraction of paired batches for which conditional LoRA has lower JS. The final row for each model gives the 95% bootstrap confidence interval for the paired difference $\Delta = \mathrm{JS}_{\mathrm{LoRA}} - \mathrm{JS}_{\mathrm{SSoT}}$. Conditional LoRA wins all $20/20$ batches on each of the four non-trivial settings for both models. The Llama two-choice interval includes zero and is treated as a statistical tie.

C.2 Temperature Sensitivity of the Prompting Baselines

The main benchmark in Table 2 uses the pinned decoding protocol reported in Appendix B, including each model's primary temperature. Because temperature can itself reshape the empirical answer distribution, we additionally test whether prompting baselines can close the gap simply by choosing a different temperature. On Gemma-2-9B, Llama-3.1-8B, and Qwen2.5-7B, we sweep $T \in \{0.6, 0.7, 1.0, 1.3\}$ for naive prompting and SSoT on all five PIF settings.

This sweep is a post-hoc sensitivity analysis rather than the primary benchmark: each temperature is evaluated with a single $K{=}100$ batch, and selecting the minimum over four temperatures favors the prompting baselines. We therefore use the independently repeated $N_b{=}20$ results in Table 2 for the main comparison.

Result. Temperature tuning can substantially improve prompting on the simplest binary setting, but it does not remove the gap on the harder targets. Across the twelve non-trivial model–setting cells, the best temperature tested for either prompting method remains worse than conditional LoRA in all twelve. Across all fifteen cells, conditional LoRA is better in $13/15$; the two exceptions are the uniform two-choice settings for Gemma-2 and Llama, where a post-hoc selected single prompting batch happens to attain lower JS. Thus, the large prompting errors on skewed and larger action spaces are not explained by the particular temperature used in the main benchmark (Table 9).

Model	Method	2-ch.	bias-2	3-ch.	bias-3	bias-9
`gemma-2-9b-it`	naive (primary $T{=}1.0$)	23.3	117.3	164.7	117.3	300.2
	naive (best T)	$1.8_{(1.3)}$	$28.2_{(1.3)}$	$77.3_{(1.3)}$	$117.3_{(0.6)}$	$278.2_{(1.3)}$
	SSoT (primary $T{=}1.0$)	6.1	25.1	6.1	25.8	32.7
	SSoT (best T)	$0.1_{(0.6)}$	$19.0_{(1.3)}$	$5.6_{(0.7)}$	$10.4_{(1.3)}$	$23.1_{(1.3)}$
	conditional LoRA ($N_b{=}20$ mean)	1.3	1.7	2.2	2.6	11.8
`llama-3.1-8b-it`	naive (primary $T{=}0.6$)	3.7	315.3	151.9	369.2	550.6
	naive (best T)	$0.8_{(1.0)}$	$248.0_{(1.0)}$	$119.0_{(1.3)}$	$325.3_{(1.0)}$	$283.3_{(1.0)}$
	SSoT (primary $T{=}0.6$)	4.1	76.4	38.0	199.3	402.2
	SSoT (best T)	$0.9_{(1.3)}$	$70.8_{(0.7)}$	$35.7_{(1.0)}$	$181.8_{(0.7)}$	$276.9_{(1.3)}$
	conditional LoRA ($N_b{=}20$ mean)	1.3	1.4	2.3	2.4	11.9
`qwen2.5-7b-it`	naive (primary $T{=}0.7$)	215.8	342.0	318.3	83.1	300.2
	naive (best T)	$193.2_{(1.3)}$	$205.0_{(1.3)}$	$318.3_{(0.6)}$	$75.5_{(1.3)}$	$300.2_{(0.6)}$
	SSoT (primary $T{=}0.7$)	152.7	172.4	213.3	327.3	258.3
	SSoT (best T)	$101.7_{(1.0)}$	$74.1_{(1.3)}$	$100.0_{(1.3)}$	$216.6_{(1.0)}$	$175.7_{(1.3)}$
	conditional LoRA ($N_b{=}20$ mean)	1.2	1.5	2.1	2.5	11.8

Table 9: **Temperature sensitivity of the prompting baselines.** JS divergence to the target ($\times 10^{-3}$). For naive prompting and SSoT, each temperature is evaluated with one $K{=}100$ batch; subscripts in the best-T rows give the selected temperature from $T \in \{0.6, 0.7, 1.0, 1.3\}$. Conditional-LoRA values are the corresponding $N_b{=}20$ means from the repeated evaluation in Table 2 and are shown only as a reference. Because best-T is selected post hoc from single batches, this table is a sensitivity analysis rather than the primary statistical comparison.

D MECHANISM REPLICATION AND ADDITIONAL CONTROL ANALYSES

We repeat the mechanistic chain of §4.3–§5.2 on `Llama-3.1-8B-Instruct` using public Llama Scope residual SAEs (`fnlp/Llama3_1-8B-Base-LXR-8x`, JumpReLU, d_{sae}=32768, dataset-wise input normalization). Because these SAEs were trained on the base model, applying them to the instruct checkpoint also tests base→instruct transfer of the feature basis.

(1) SAE latent. Across blocks $\{7, 11, 15, 19, 23, 27\}$, we find highly selective random-choice features (random-vs-deterministic AUC 0.997–1.000). The cleanest, `L11F9418`, fires on 100% of random requests (including 100% of implicit requests), 14% of deterministic prompts, and 0% of "random"-mention controls (AUC 0.998), reproducing the concept-not-token signature of Gemma's `L20F2001`.

(2) Switch vs. writer. Gradient attribution through the block-11 SAE basis again separates the mode feature from the distribution-shaping direction. Activating the random-choice feature pushes probability away from the valid answer set, whereas distinct, mostly inactive latents carry positive uniformity attribution. Steering the attribution-selected writer raises random entropy from 1.83 to 2.55 while increasing valid mass from 0.31 to 0.46. The mode reader remains near 1.9 bits, and a norm-matched random direction reduces entropy and valid mass. This reproduces the reader–writer dissociation of §4.4.

(3) Rank-1 edit. Using the same train/validation/test protocol as §5.2, the hand-built mode-gated edit raises held-out random entropy from 2.03 to 2.23 and valid mass from 0.20 to 0.29, while maintaining deterministic accuracy at 0.95. This recovers 22% of the matched rank-8 LoRA entropy gain and 15% of its valid-mass gain. The matched LoRA reaches entropy 2.94 and mass 0.78; Table 1 reports seed means and therefore differs slightly. Removing the gate collapses deterministic accuracy to 0.14, while a random reader direction (2.04 bits) and a wrong-layer edit (2.03 bits) are inert.

Recovery is smaller than on Gemma, partly because Llama begins with higher test-domain entropy (2.03 vs. 1.48), leaving less headroom to the trained ceiling.

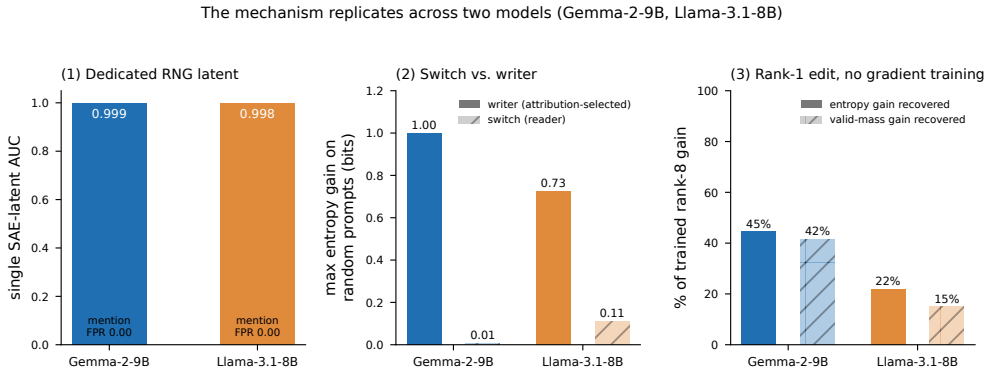

Figure 14: **The reader–writer mechanism replicates across Gemma-2-9B and Llama-3.1-8B.** (1) Both models contain a highly selective SAE feature for the random-choice state (AUC ≈ 0.999) that does not fire on "random"-mention controls. (2) Steering the attribution-selected writer increases entropy (1.00/0.73 bits for Gemma/Llama), whereas steering the mode-reader direction has little effect (0.01/0.11 bits). (3) The hand-built rank-1 reader×writer edit, with no task-specific training, recovers 45%/22% of the full LoRA entropy gain and 42%/15% of its valid-mass gain.

D.1 ISOLATING THE TRAINED ADAPTER'S BRANCH

The block-input ablation (Fig. 15) projects $\hat{u}_{\text{mode}}$ out of the residual before the adapted block, affecting both the base path Wh and the adapter path BAh. To test whether the *adapter itself* depends on this direction, we instead project it out only inside the adapter term, $Wh + BAP_{\perp}h$, at the residual-reading projections (gate/up) of every adapted block, leaving the base path unchanged.

Table 10 reports the full comparison. Removing the mode direction from the LoRA branch eliminates only 2.1% of the adapter's entropy gain, while a random-direction control removes 0.0%. In contrast, projecting the direction globally removes 52% of the gain, but this largely reflects damage to the base path: applying the same projection to the base model alone reduces valid mass from 0.48 to 0.14.

Thus, the trained rank-8 adapter does not substantially depend on the localized mode direction within its own branch and instead implements the conditional behavior through a more distributed solution. The training-free intervention of §5.2, rather than the trained adapter, provides the causal evidence for the localized mechanism.

Condition	test random H	% adapter gain removed
base model	1.48	—
adapted (rank-8)	3.10	0%
global projection (base + LoRA)	2.26	52%
LoRA-only projection (base untouched)	3.07	**2.1%**
LoRA-only, random direction	3.10	0.0%

Table 10: LoRA-branch-only vs. global mode ablation (Gemma-2-9B, locked test domains). Held-out random entropy (bits) and fraction of the adapter's gain over base removed.

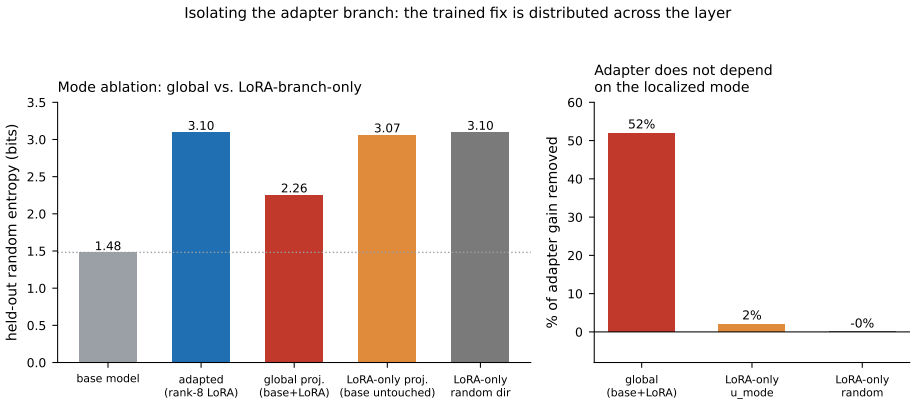

Figure 15: **The trained rank-8 adapter is mode-conditional but does not substantially depend on the localized mode direction.** Held-out random entropy under five conditions. Projecting $\hat{u}_{\text{mode}}$ out of the residual at every adapted block's *input* (global) removes 52% of the gain ($3.10 \to 2.26$), but this also corrupts the base path (base valid mass $0.48 \to 0.14$). Projecting it out of *only the adapter branch* (base path untouched, $Wh + BA\hat{P}_{\perp}h$) removes just 2.1% ($3.10 \to 3.07$); a random branch direction removes none. The adapter therefore reaches its behavior through a more distributed solution and does not substantially depend on the localized mode direction within its own branch. Deterministic accuracy preserved throughout; Gemma-2-9B, locked test domains.

D.2 Which Direction Calibrates? A Mass-Aware Writer Taxonomy

The branch analysis (Appendix D.1) shows that the trained adapter implements its correction through a distributed solution. We therefore ask whether any single rank-1 writer direction can reproduce that behavior, and whether such a direction can be obtained from the base model alone. Every candidate uses the same gated rank-1 intervention as §5.2, with the same reader $\hat{u}_{\text{mode}}$, block ℓ=31, and matched direction norm. Its strength α is selected on validation domains subject to valid-answer mass ≥ 0.60 and deterministic $p_{\text{modal}} \geq 0.85$, after which random entropy is maximized.

The mass constraint is important when comparing many candidate directions. Because random-choice entropy is computed after renormalizing over valid answers, a large intervention can appear highly uniform while moving probability outside the valid set. Under entropy-only selection, even random directions can therefore show near-complete apparent recovery. With the mass guard, the

ten-direction random control falls to 9% recovery at mass 0.49 (Fig. 16, Table 11). This issue does not affect the headline edit of §5.2, where the writer is fixed in advance and only its strength is selected.

Two directions reach high entropy while retaining high valid mass. The trained adapter's mean activation shift $\bar{d} = \mathbb{E}_{\text{rand}}[h_{\text{LoRA}} - h_{\text{base}}]$ at block 31 recovers 92% of the trained entropy gain at valid mass 0.84, while the attribution-selected SAE latent reaches 64% at mass 0.66. In contrast, all base-model-only directions we tested remain near the base-mass regime, including the mode reader, entropy difference-in-means, instruction contrast, entropy gradient, and gradient of the mass-aware objective (Table 11). Thus, although the base model clearly represents the random-choice state, we do not find a base-derived single direction that both spreads probability across valid answers and keeps the probability mass on them.

The strong mean-shift result is model-specific. On Llama-3.1-8B, the same analysis peaks at only 34% recovery with valid mass 0.38, consistent with a more distributed trained solution. We therefore treat $\bar{d}$ as a Gemma-specific writer extracted post hoc from a calibration-trained adapter, not as a general training-free method. Finding a base-model-only, model-general writer remains open.

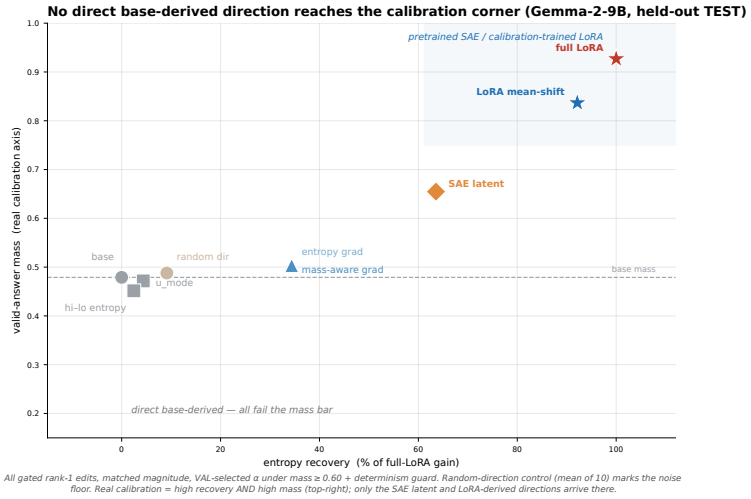

Figure 16: **No base-derived direction reaches the calibration corner.** Each point is a gated rank-1 writer at block 31, evaluated on the locked Gemma-2-9B test domains after mass-aware strength selection. The x-axis shows recovery of the full-LoRA entropy gain and the y-axis valid-answer mass. Base-derived directions remain near the base-mass regime, whereas the pretrained-SAE writer and the LoRA-derived mean shift $\bar{d}$ achieve both higher entropy recovery and higher valid mass.

D.3 A Low-Rank Installer: LoReFT vs. the LoRA Calibrator

This appendix gives the full details behind §5.4. Figure 17 summarizes the held-out comparison between rank-4 LoReFT and the rank-8 LoRA calibrator. We use the LoReFT intervention

$$\Phi(h) = h + R^{\top}(W(Rh) + b - Rh)\,,$$

where $R \in \mathbb{R}^{r\times d}$ is row-orthonormal (re-orthonormalized by QR after each step), $W \in \mathbb{R}^{r\times r}$, and $b \in \mathbb{R}^{r}$. Only R, W, and b are trained; the base model remains frozen. The intervention is applied to the residual stream at the same late blocks as our LoRA.

Objective, data, train/held-out split, and evaluation are identical to the rank-8 LoRA of §5.1. On random prompts, we optimize

$$\mathcal{L}_{\text{rand}} = \mathrm{KL}(q \,\|\, p_{\text{target}}) - \lambda_{\text{mass}} \log m_{\text{valid}},$$

together with a $\mathrm{KL}(p_{\text{base}} \,\|\, p_{\text{new}})$ preservation term on deterministic and mention prompts. Only the trainable module differs.

Source	Direction	recovery	valid mass
—	random-direction control (mean of 10)	9%	0.49
base-derived	mode reader $\hat{u}_{\text{mode}}$	4%	0.47
base-derived	high−low entropy diff-in-means	2%	0.45
base-derived	instruction contrast (uniform − bare)	2%	0.47
base-derived	entropy gradient	34%	0.50
base-derived	mass-aware objective gradient (λ^*=0.5)	31%	0.42
pretrained SAE	SAE latent `L31F7399` decoder row	64%	0.66
LoRA-derived	LoRA mean activation shift $\bar{d}$	**92%**	**0.84**

Table 11: **Mass-aware writer comparison on Gemma-2-9B.** Recovery is the fraction of the full-LoRA entropy gain, $(H - 1.48)/(3.10 - 1.48)$, and mass is valid-answer mass (base 0.48, LoRA ceiling 0.93). Base-derived directions use only the base model; the pretrained-SAE row uses a decoder row from a public reconstruction SAE; the LoRA-derived row is extracted post hoc from a calibration-trained adapter. The mass and p_{modal} constraints are applied on the validation domains during α selection; the masses reported here are measured afterwards on the locked test domains, where the unedited base model has valid mass 0.48. Thus, test mass below 0.60 does not imply a violation of the validation constraint. If no α satisfies the validation constraints, the smallest grid value α=2 is used.

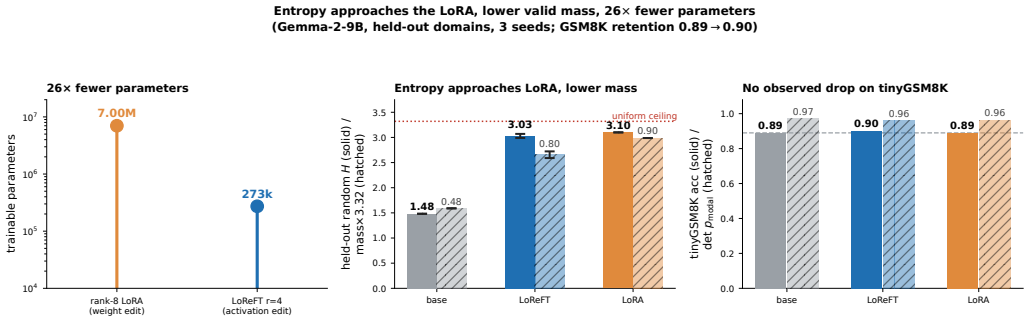

Figure 17: **Low-rank LoReFT approaches LoRA with 26-fold fewer trainable parameters.** On Gemma-2-9B held-out domains, rank-4 LoReFT reaches H $= 3.03 \pm 0.04$ at valid mass 0.80 ± 0.02, versus H $= 3.10$ and mass 0.90 for rank-8 LoRA, with no observed tinyGSM8K degradation. Entropy and mass are averaged over three seeds; the capability panel uses the seed-0 adapter for each method. Additional rank sweeps and Llama results are reported below.

On Gemma-2-9B, where the mechanism is derived, rank-4 LoReFT reaches the finite-sample PRNG calibration regime on all five PIF settings, with mean empirical JS divergence ($\times 10^{-3}$) of 1.3/1.3/2.5/2.0/11.1 for 2-choice, biased-2, 3-choice, biased-3, and biased-9, respectively (Table 2). Transfer of the same fixed recipe across models is less uniform, as summarized in Table 2.

Activation-delta basis (ADB): estimating the rank before LoReFT. We first estimate the relevant intervention dimension independently of LoReFT, using only training domains. At block 31, we collect the activation changes induced by the trained rank-8 LoRA,

$$D = \begin{bmatrix} \Delta h_1 \\ \vdots \\ \Delta h_n \end{bmatrix}, \qquad \Delta h_i = h_{\text{LoRA},i} - h_{\text{base},i},$$

over random-choice prompts, and compute the SVD of D (Fig. 18a).

The spectrum is strongly concentrated: the top component contains 73% of the energy, while cumulative energy reaches 95% by r=8 and 97% by r=11 (r_{95}=8, r_{97}=11).

Reconstructing the writer from the top-K directions and installing it as a gated rank-1 edit already recovers ∼87% of the trained entropy gain at K=1, with little additional improvement through K=32 (Fig. 18b). Because this reconstruction score measures entropy alone and ignores valid-

answer mass, we use the activation-delta energy spectrum as the rank signal. It suggests that a learned representation intervention at block 31 should saturate at roughly the 8–11 dimensional scale, which we test below with a matched single-block LoReFT sweep.

Does the measured dimension simply reflect LoRA rank? Because the ADB matrix is estimated from a rank-8 LoRA, the measured dimension could in principle be an artifact of the adapter's architectural rank. We therefore repeat the analysis using calibration LoRAs trained at ranks $4, 8, 16,$ and 32, with the same objective, data, and split and comparable held-out behavior. If the activation dimension simply tracked LoRA rank, r_{95} should increase with adapter rank. Instead, it remains nearly constant: $r_{95} = 8, 8, 8, 6$ and $r_{97} = 11, 11, 12, 9$ for ranks $4, 8, 16, 32$, respectively (Fig. 19). Thus, even substantially higher-rank adapters produce activation changes concentrated in approximately the same $\sim$10-dimensional subspace.

The four adapters also reach comparable behavioral endpoints (Table 12), reducing the possibility that the stable activation dimension is explained simply by large performance differences between runs. Comparable behavior does not, however, imply identical internal solutions.

LoRA rank	H (held-out)	valid mass	det. p_{modal}	block-31 r_{95}
4	3.12	0.92	0.96	8
8	3.10	0.90	0.96	8
16	3.12	0.92	0.97	8
32	3.12	0.87	0.96	6

Table 12: Calibration LoRAs at ranks $4, 8, 16, 32$ reach a matched behavioral endpoint (Gemma-2-9B, held-out domains, single seed, 300 steps, identical objective/data/split). Similar entropy *and* valid mass *and* preserved determinism mean the adapters reach comparable behavioral endpoints, making the activation-dimension comparison less confounded by gross performance differences; this supports the conclusion that the measured dimension does not track LoRA training rank, shown in Figure 19.

Does the concentration hold at the per-example level? The gated rank-1 edit uses the mean writer $\bar{d} = \mathbb{E}_i[\Delta h_i]$, so its success at $K{=}1$ could arise even if the individual, prompt-conditioned activation changes were high-rank. We test this directly by fitting the top-K subspace on one random half of the training prompts and reconstructing the individual deltas on the other half. At $K{=}8$, the subspace achieves mean per-example cosine 0.94 and explains 88% of held-out delta variance ($0.83/0.70$ at $K{=}1$ and $0.95/0.90$ at $K{=}16$). Thus, the per-example deltas—not only their mean or aggregate spectrum—are concentrated in a low-dimensional subspace.

The structure also transfers across domains. Fitting the subspace on the training domains and reconstructing deltas on the held-out domains (die/month/color) gives cosine 0.86 and explained variance 0.74 at $K{=}8$ ($0.78/0.62$ at $K{=}1$). The concentration therefore generalizes beyond the domains used to estimate the subspace, although less tightly than within-domain.

Matched single-block test at block 31. The activation-delta spectrum is measured at block 31, whereas the LoReFT intervention of Table 13 acts at all 19 late blocks. Rank 4 in that experiment therefore means four intervention dimensions *per block*, so its rank frontier is not a direct test of the block-31 estimate. We run the matched experiment by restricting LoReFT to block 31 and sweeping

$$r \in \{1, 2, 4, 8, 11, 16, 32\}.$$

Figure 20 reports the sweep, with three seeds at $r \in \{4, 8, 11, 16, 32\}$ (mean $\pm$ sd).

The results are consistent with saturation around the scale predicted by the activation-delta spectrum, but do not show a sharp boundary. Held-out valid mass is $0.63{\pm}0.01$ at $r{=}4$, $0.62{\pm}0.03$ at $r{=}8$, and $0.69{\pm}0.01$ at $r{=}11$, with no consistent improvement at $r{=}16$ ($0.72{\pm}0.03$) or $r{=}32$ ($0.68{\pm}0.05$). The $r{=}11$, 16, and 32 results substantially overlap under the observed seed variation.

A single block is nevertheless insufficient to recover the full policy: block 31 saturates near valid mass $\sim$0.70, below the 19-block LoReFT (0.80) and rank-8 LoRA (0.90). The calibration policy is therefore distributed across depth. We estimate the activation-delta dimension only at block 31; the

multi-layer intervention can succeed at low *per-block* rank by composing low-rank changes across many blocks. Accordingly, the 8–11 estimate concerns the intervention rank at block 31, not the dimensionality of the full calibration policy or of every late block.

Multi-layer LoReFT rank frontier. We separately sweep the multi-layer LoReFT rank through r=32 (Table 13, Fig. 21). Held-out valid mass is $0.71/0.78/0.80/0.72/0.83/0.80$ and entropy is $2.97/2.91/3.03/2.98/2.97/3.05$ at r=$1/2/4/8/16/32$, respectively. Neither metric improves monotonically with rank, and there is no consistent gain above low rank. The apparent r=8 mass dip in the single-seed run (0.65) is within seed variation: the three-seed mean is 0.72 ± 0.06, overlapping the r=4 result.

Rank-4 LoReFT therefore approaches the rank-8 LoRA on Gemma-2-9B: held-out entropy $3.03\pm.04$ versus 3.10, valid mass $0.80\pm.02$ versus 0.90, and GSM8K 0.90 versus 0.89, while preserving deterministic $p_{\text{modal}} = 0.96$. It uses 272,764 trainable parameters rather than ∼7M, a 26× reduction. Even rank 1 (68,134 parameters, 103× fewer) reaches H=2.97 at valid mass 0.71.

Rank-4 performance also transfers to Llama-3.1-8B (246,060 parameters; H 2.03 → 2.93, valid mass 0.72), but the full rank sweep was run only on Gemma-2-9B. The saturation claim is therefore Gemma-specific; Llama supports only the rank-4 result. LoReFT's general parameter efficiency is established by prior work (Wu et al., 2024); the contribution here is that analysis of the trained policy predicts a low *per-block* intervention-rank scale, and the matched block-31 sweep is consistent with that prediction.

This claim has three scope limits. The activation-delta subspace analysis concerns the *uniform-calibration* policy on Gemma-2-9B, so we do not claim that the biased-target or prompt-conditioned distribution installers share the same rank. The full policy is distributed across depth, with a single block capping near valid mass ∼0.70. Finally, only r=4 and r=8 have three-seed estimates in the multi-layer sweep; the remaining multi-layer ranks are single-seed.

Model	Method	params	H	valid mass	GSM8K	param ratio
gemma-2-9b-it	rank-8 LoRA	∼7M	3.10	0.90	0.89	1×
gemma-2-9b-it	LoReFT r=1	68,134	2.97	0.71	—	103×
gemma-2-9b-it	LoReFT r=2	136,306	2.91	0.78	—	51×
gemma-2-9b-it	**LoReFT** r=4	272,764	**3.03±.04**	**0.80±.02**	**0.90**	**26×**
gemma-2-9b-it	LoReFT r=8	546,136	2.98±.05	0.72±.06	—	13×
gemma-2-9b-it	LoReFT r=16	1,094,704	2.97	0.83	—	6×
gemma-2-9b-it	LoReFT r=32	2,199,136	3.05	0.80	—	3×
llama-3.1-8b-it	rank-8 LoRA	∼6M	2.94	0.78	—	1×
llama-3.1-8b-it	LoReFT r=4	246,060	2.93	0.72	—	∼25×

Table 13: LoReFT rank sweep vs. rank-8 LoRA at matched objective/data/split (held-out domains). Gemma r=4 and r=8 are mean±sd over three seeds; all other rows are single-seed. GSM8K is greedy tinyGSM8K accuracy (base 0.89); the Llama rank-8 LoRA reference is a single measured adapter run (Appendix D.1); its mass 0.78 is one draw from the n=3 mean 0.75 ± 0.05 reported in Table 1. The Gemma LoReFT points preserve deterministic $p_{\text{modal}} \geq 0.954$; the Llama r=4 point is 0.92 (Llama base 0.93). Neither entropy nor valid mass improves monotonically with rank.

Cross-model robustness of rank-4 LoReFT. We next test rank-4 LoReFT on all five PIF settings across all five models using a common objective and no per-cell retuning. Because an initial reproduction selected checkpoints using TV after renormalizing over valid answers, it could prefer a checkpoint with negligible valid-answer mass. We therefore use the same pre-existing mass-aware validation rule in every cell: checkpoints are eligible only when valid-answer mass is at least 0.5, after which validation TV selects the checkpoint. The training objective, KL direction, rank, learning rate, steps, seeds, data splits, answer mappings, targets, and intervention locations are otherwise unchanged.

The corrected selector changes only one of the 75 seed-level checkpoint choices relative to the mass-blind rule when both are recomputed on the same training traces. This occurs for Qwen2.5 two-choice, where the mass-blind rule would select a step with only 0.06% valid-answer mass because its renormalized conditional distribution appears well shaped. The mass-aware rule instead selects a

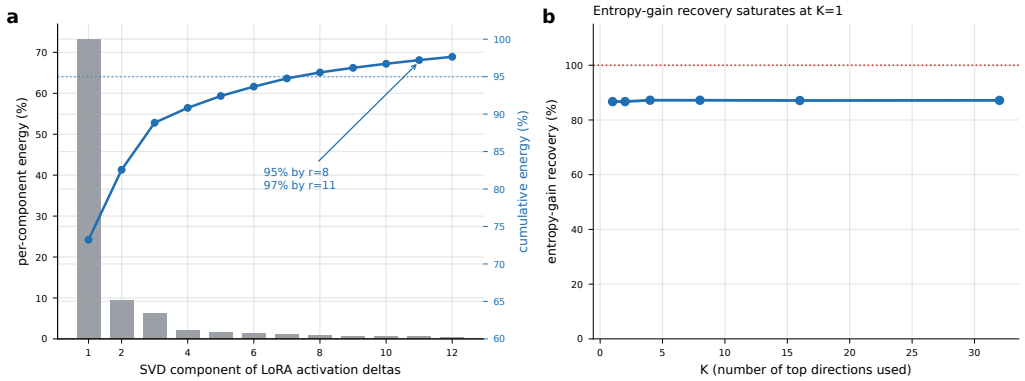

Figure 18: **LoRA-induced calibration deltas occupy a $\sim$10-dimensional activation subspace** (Gemma-2-9B, block 31, training domains). **(a)** SVD of the LoRA-induced activation deltas: the top component holds 73% of the energy; cumulative energy reaches 95% by r=8 and 97% by r=11. **(b)** Reconstructing the writer from the top-K directions and installing it as a gated rank-1 edit recovers $\sim$87% of the trained *entropy gain* at K=1 and does not increase through K=32 (this top-K score tracks entropy gain alone and ignores valid mass, so it falls short of a full behavioral measure). The falsifiable rank signal is the activation-delta energy dimension (r_{95}=8, r_{97}=11), predicted by the trained-policy analysis, which motivates the expected rank scale of a learned representation intervention; a single mean direction suffices for the entropy statistic on its own; the corresponding saturation prediction is tested at block 31 in Figure 20.

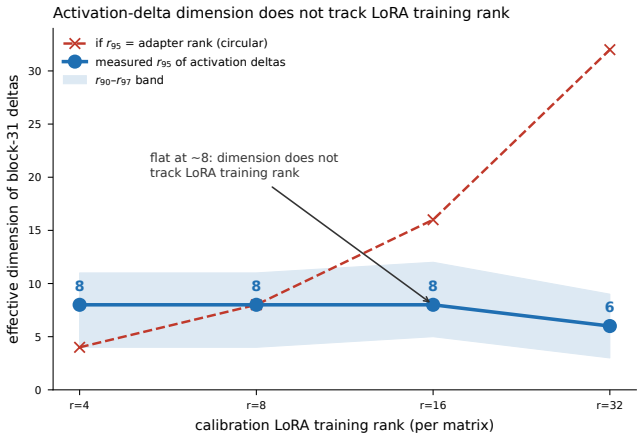

Figure 19: **The activation-delta dimension does not track the adapter's training rank** (Gemma-2-9B, block 31). Re-estimating the ADB spectrum from calibration LoRAs trained at ranks $4, 8, 16, 32$ (each reaching the same entropy ceiling) leaves r_{95} flat at $\sim$8 (dashed red: what a circular "you found the adapter's rank" account would predict). A rank-32 adapter, eight times the per-matrix capacity, yields a subspace of the same $\sim$10 dimensions, so the dimension is not explained by the LoRA training rank. Across four single-seed trainings this argues against the circular rank-tracking reading; it does not rule out every shared property of LoRA training.

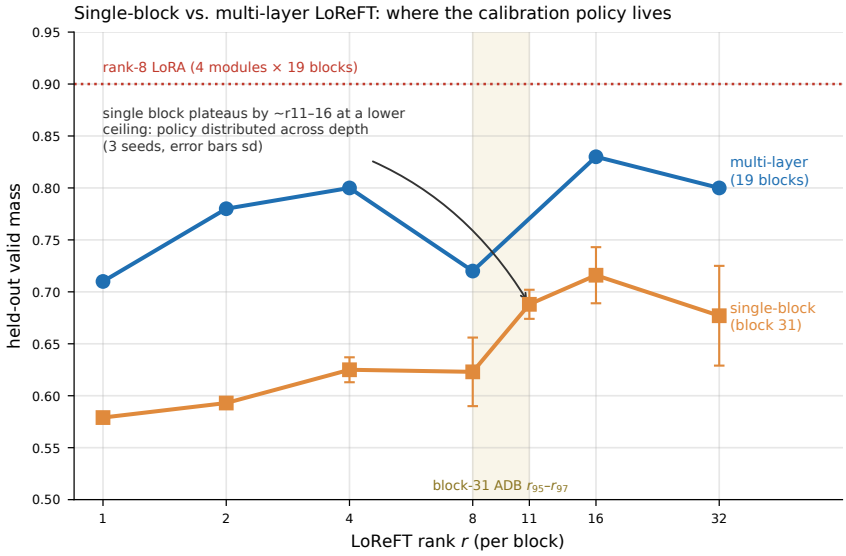

Figure 20: **Single-block vs. multi-layer LoReFT: the calibration policy is distributed across depth** (Gemma-2-9B, held-out domains). Restricting LoReFT to block 31, where the ADB spectrum is measured, is consistent with saturation by roughly $r{\approx}11$, the order of magnitude predicted by the $r_{95\text{–}97}{=}8$–11 estimate (shaded), with no sharp boundary within it, but at a lower ceiling (valid mass $\sim$0.70) than the 19-block LoReFT (0.80) or the LoRA (0.90). A single block is not enough; the multi-layer edit composes $\sim$19 low-rank per-block interventions, which is why low *per-block* rank suffices.

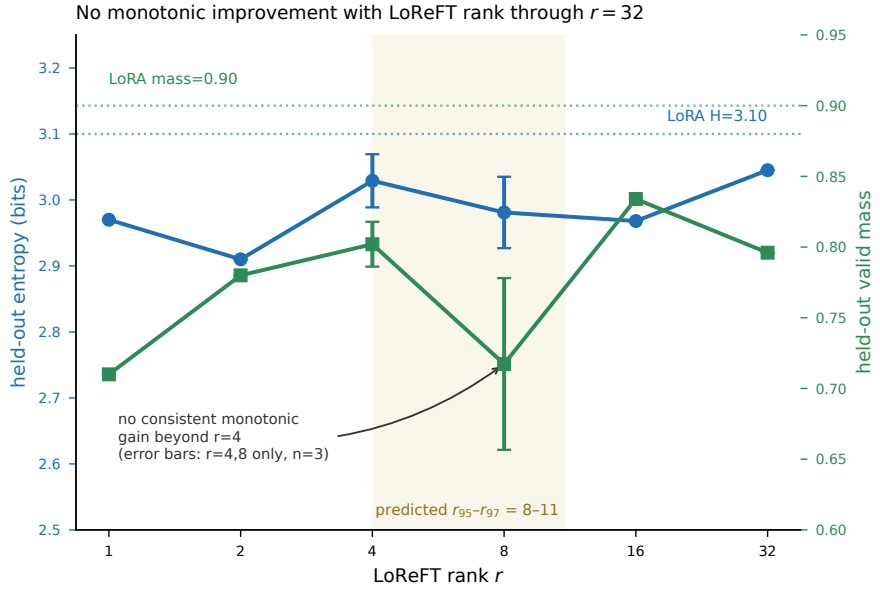

Figure 21: **LoReFT rank frontier** (Gemma-2-9B, held-out domains; $n{=}3$ at $r{=}4, 8$ with sd error bars, $n{=}1$ at all other ranks). Even a rank-1 intervention (68k parameters, 103$\times$ fewer than the LoRA) approaches the LoRA's calibration entropy, and neither held-out entropy nor valid mass improves monotonically with rank through $r{=}32$. Spread across 19 blocks, then, very low rank already suffices; this sweep does not by itself confirm the $\sim$10-dimensional estimate of the ADB analysis (Fig. 18), for which the matched test is the block-31 sweep of Figure 20. The shaded band marks the ADB $r_{95\text{–}97}$ range for reference. The head-to-head summary is shown in Figure 17 in the main text.

high-mass checkpoint and exposes poor calibration. Thus, the correction prevents hollow low-mass solutions rather than systematically changing the LoReFT training outcome.

Under the fixed recipe, 20/25 model–target cells do not exhibit the failure modes observed in the remaining five. The exceptions are Llama biased-3, Qwen2.5 two-choice and biased-3, and Qwen3 biased-2 and biased-3. In particular, all three biased-3 failures persist after the checkpoint-selection correction. These failures retain substantial valid mass but mis-shape the distribution, typically suppressing the smallest target component; this pattern is consistent with, but does not by itself establish, mode-seeking behavior under the reverse-KL objective (Table 2).

D.4 Selection controls for the rank-1 writer

The rank-1 reader $\hat{u}_{\text{mode}}$ and intervention strength α are selected under the train/validation/test discipline of §5.2. The SAE writer was originally identified by uniformity attribution over random prompts from all domains. To test whether this introduced test-domain information, we repeat the *entire writer-selection procedure* using training domains only. We recompute uniformity attribution over the complete block-31 SAE latent set using only training-domain random prompts, then apply the same rule: select the highest-uniformity-attribution latent with positive mass attribution and firing rate below 0.05. This train-only search again selects `L31F7399`. Installed with the same reader, it reaches $H = 2.23$ on the locked test domains at α^*=4, reproducing the reported effect. Thus, although the original discovery was retrospective, the selected writer does not depend on the test domains.

We next test whether the selected reader and writer are exceptional relative to matched controls. Holding the intervention pipeline fixed, we evaluate 50 random unit reader directions paired with the fixed writer and 50 frequency-matched SAE latents paired with the fixed reader, all at α^*=4 on the locked test domains. The selected reader×writer reaches $H = 2.23$, exceeding every control draw: the reader-null distribution has mean 1.47 and maximum 1.60, while the writer-null distribution has mean 1.52 and maximum 1.83 (base $H = 1.48$). None of the 50 controls in either null matches or exceeds the selected direction.

The small difference from the $H = 2.20$ reported in §5.2 arises because this control script estimates the reader from the residual stream after block 31 rather than before it. The resulting reader is highly aligned with the headline reader (cosine 0.94) but has a slightly larger contrast norm (262.0 rather than 252.4), yielding a slightly stronger effective edit at the same α^*=4.

D.5 Support Size and Target Shape of the Hand-Built Rank-1 Edit

The hand-built reader–writer intervention in Sec. 5.2 is selected for increasing choice diffuseness, so we test how its behavior changes with both support size and target shape. On Gemma-2-9B we hold the train-domain reader, block-31 SAE writer, and train/validation/test discipline fixed, and evaluate supports of $n \in \{2, 4, 8, 16\}$ under uniform, increasing-ramp, and geometric targets. For each cell, α is selected from $\{0.5, 1, 2, 4, 8, 16\}$ on validation prompts subject to deterministic accuracy ≥ 0.90. The same value, $\alpha^* = 4$, is selected in all 12 cells (Table 14).

The direction of the effect is target-independent: for a fixed support, the edit raises entropy and generally valid-answer mass regardless of whether the requested target is uniform or skewed. For example, at $n = 16$ entropy rises from 3.08 to 3.50 bits and valid mass from 0.755 to 0.830. This helps the uniform target substantially (TV $0.444 \rightarrow 0.318$) but only weakly improves the strongly geometric target (TV $0.854 \rightarrow 0.827$). Thus, the hand-built writer is best characterized as a *diffuseness/uniformity* direction, not a general arbitrary-target distribution writer; arbitrary target shapes require the learned controllers evaluated in Sec. 6.2.

D.6 Stochastic vs. Arbitrary: What Does the Direction Encode?

Does the mode direction specifically encode stochastic sampling, or a broader open-choice state? We construct 162 prompts by deterministically expanding author-written templates across nine domains and six semantic classes: personal preference, arbitrary free choice ("give any X"), epistemic uncertainty ("guess the one I picked"), a deterministic pseudo-random rule, a random process with fixed output, and genuinely stochastic sampling.

Support n	Target	Base TV	Rank-1 TV
2	uniform	0.299	0.274
2	ramp	0.133	0.107
2	geometric	0.445	0.419
4	uniform	0.332	0.313
4	ramp	0.232	0.192
4	geometric	0.499	0.489
8	uniform	0.287	0.244
8	ramp	0.249	0.225
8	geometric	0.722	0.697
16	uniform	0.444	0.318
16	ramp	0.375	0.276
16	geometric	0.854	0.827

Table 14: **Support-size and target-shape sweep for the hand-built rank-1 edit.** TV is measured against the requested target distribution on the locked test prompts. The same $\alpha^* = 4$ is selected in every cell. The intervention improves TV across all tested cells, with the largest gains on uniform or moderately diffuse targets; strongly geometric targets remain far from the requested distribution.

We project each prompt's best-layer activation onto the frozen mode direction estimated from the original training data. The direction only weakly distinguishes genuinely stochastic output from the other five classes (AUC 0.61 for both models). In particular, it is near chance for stochastic sampling versus arbitrary free choice (Gemma 0.50, Llama 0.57) and versus random-process prompts with fixed output (0.51/0.58). Separation is stronger for deterministic pseudo-random rules (0.77/0.70) and uncertainty (0.68). Consistently, the class-wise projections place arbitrary and stochastic choice in the same high-activation regime, while preference and uncertainty are lower (Fig. 22).

Independent probes trained separately for the semantic classes achieve AUC 0.96–1.00, but these results are confounded by class-specific template vocabulary, as in the lexical controls above. The frozen-direction projection is therefore the more informative test of what the previously identified representation encodes. We conclude that the direction reflects a broader random-or-arbitrary choice state rather than stochasticity itself.

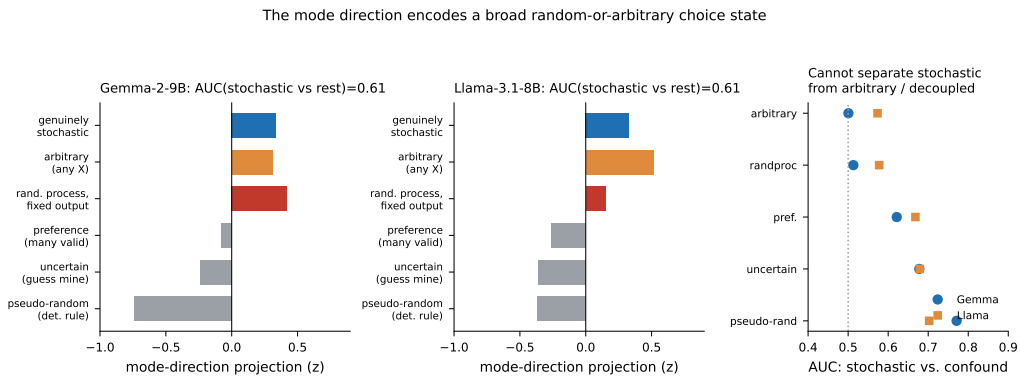

Figure 22: **The mode direction reflects a broad random-or-arbitrary choice state.** Stochastic and arbitrary-choice prompts occupy the same high-projection regime, and the frozen direction cannot reliably distinguish stochastic sampling from arbitrary free choice or random-process prompts with fixed output.

D.7 Reliability of the SAE-Based Analysis

Two findings rely directly on public sparse autoencoders: the selective choice-state feature of Sec. 4.3 and the attribution-selected entropy writer, which is an SAE decoder direction. SAEs

have documented failure modes, including feature splitting and *absorption*, where a conceptual feature is distributed across or absorbed into other features (Chanin et al., 2025); *interpretability illusions*, where common feature-quality metrics can fail to distinguish trained from randomly initialized transformers (Heap et al., 2026); and out-of-distribution brittleness, where apparently clean features generalize poorly beyond their original distribution (Heindrich et al., 2025). These caveats therefore apply directly to our SAE-specific interpretations.

Several aspects of the analysis reduce, but do not eliminate, this dependence. First, the underlying choice-state representation is established directly in the residual stream using linear probes, independently of any SAE, with leave-one-domain-out and implicit-prompt generalization. The selective SAE feature is therefore corroborating evidence rather than the sole evidence for the state. Second, the SAE→attribution→rank-1 intervention chain replicates qualitatively on Llama-3.1-8B using independently trained public Llama Scope SAEs (Appendix D); we claim replication of the mechanistic pattern, not of particular latent indices. Third, the SAE-dependent causal results are tested through interventions and matched controls, including random-reader, random-writer, and wrong-layer nulls. Their behavioral effects therefore do not require the selected SAE features to be perfectly monosemantic.

SAE-free writer test. The reader itself is SAE-independent, so we additionally ask whether the writer can be recovered directly from base-model gradients. On Qwen2.5 and Qwen3, we construct (i) the gradient of valid-answer entropy with respect to the residual stream and (ii) a mass-aware gradient of entropy plus $0.5 \log(\text{valid mass})$. We install each as the same gated rank-1 writer and select $\alpha \in \{0.5, 1, 2, 4, 8\}$ subject to both $p_{\text{det}} \geq 0.9\, p_{\text{det}}^{\text{base}}$ and validation valid mass at least its base value; ten random writers provide a null. Table 15 reports the resulting held-out performance.

Model	Writer	α^*	H_{test}	valid mass	entropy recovery
Qwen2.5	entropy gradient	—	—	—	no admissible α
Qwen2.5	mass-aware gradient	0.5	1.876	0.231	-7.4%
Qwen3	entropy gradient	0.5	1.370	0.180	8.5%
Qwen3	mass-aware gradient	0.5	1.219	0.274	0.4%

Table 15: **SAE-free writer directions do not reproduce the SAE-derived writer.** Entropy recovery is measured relative to each model's conditional-LoRA ceiling on the same held-out domains. Selection is mass- and preservation-aware; a dash indicates that no tested strength satisfies the constraints.

These base-model gradient directions therefore do not recover the strong SAE-derived writer effect: Qwen2.5 has no admissible entropy-gradient edit and its mass-aware direction reduces entropy, while Qwen3 recovers at most 8.5% of the trained entropy gain without improving valid mass. This does not show that an SAE-free writer is impossible, but it leaves the current hand-built writer dependent on finding a suitable SAE-derived direction.

D.8 Weight Effect vs. Decoder Effect, and the Mass-Reward Weight

Restricted-marginal vs. end-to-end generation. The main PIF benchmark evaluates trained installers by sampling from the installed valid-answer marginal, whereas naive prompting and SSoT use full autoregressive generation. To test whether the learned intervention survives this change of interface, we run a separate protocol diagnostic on Gemma-2-9B and Llama-3.1-8B across all five PIF settings. For the same mass-corrected conditional-LoRA adapter, we compare (P1) the analytic valid-answer marginal, (P2) one host-side $K{=}100$ sample from that marginal, and (P3) $K{=}100$ unconstrained free generations from the adapted model. For reference, we also run naive prompting on the base model with the target distribution stated explicitly in the prompt. This diagnostic is separate from the repeated $N_b{=}20$ benchmark in Table 2. Table 16 reports the protocol comparison.

The broader diagnostic agrees with the earlier 2×2 Rock–Paper–Scissors control shown in Figure 23: adapting the weights improves calibration under both restricted and free decoding. The new five-setting comparison makes clear, however, that restricted-marginal calibration and end-to-end generation are not numerically equivalent protocols.

Model	Setting	analytic TV	restricted TV	adapted free TV	naive-base TV	parsed (adapt/base)
Gemma-2	2-choice	0.000	0.060	0.133	0.140	98/100
	biased-2	0.000	0.060	0.068	0.300	99/100
	3-choice	0.001	0.077	0.327	0.333	97/100
	biased-3	0.001	0.050	0.140	0.300	100/100
	biased-9	0.003	0.080	0.170	0.640	100/100
Llama-3.1	2-choice	0.002	0.060	0.170	0.015	100/99
	biased-2	0.006	0.070	0.040	0.700	100/86
	3-choice	0.006	0.077	0.083	0.550	100/60
	biased-3	0.004	0.050	0.390	0.743	100/51
	biased-9	0.003	0.080	0.220	0.640	100/5

Table 16: **Protocol diagnostic: installed marginal versus full generation.** TV is measured against the requested target distribution. "Analytic" is the conditional-LoRA valid-answer marginal before sampling; "restricted" is one $K{=}100$ host-side draw from that marginal; "adapted free" is unconstrained autoregressive generation from the adapted model on the ordinary random-choice prompt; and "naive base" is the unadapted model with the target distribution stated in the prompt. The adapted model improves over this naive baseline in $9/10$ cells, with Llama two-choice as the exception. The larger gap between analytic and free-generation TV shows that downstream generation introduces additional error even when the answer-position distribution is well calibrated.

The PRNG reference is itself a distribution. At finite K, even exact draws from the target have nonzero empirical divergence. For the matched reference used in Table 2, we draw $N_b{=}20$ independent batches of $K{=}100$ directly from each target. The resulting JS divergence (mean; median; central 95% interval, all in $\times 10^{-3}$) is: 2-choice $0.98; 0.80; [0.00, 3.22]$, biased-2 $1.50; 0.96; [0.06, 3.89]$, 3-choice $2.92; 1.85; [0.10, 8.17]$, biased-3 $2.09; 1.75; [0.33, 4.39]$, and biased-9 $9.55; 9.20; [4.74, 14.46]$. We therefore refer to a finite-sample PRNG *reference interval*, not a zero-error "floor." Conditional LoRA lies inside the corresponding interval in every main-benchmark cell; Table 8 gives the sampling-aware paired comparison with SSoT.

Mass-reward weight. The valid-mass reward coefficient, denoted λ_{mass} to distinguish it from unrelated hyperparameters, differs across experiments (Table 17). The entropy/uniformity installers use $\lambda_{\text{mass}}{=}0.05$. Our initial PIF experiments used no mass term ($\lambda_{\text{mass}}{=}0$), which allowed substantial probability leakage outside the valid answer set; this setting is retained in Table 17 only as an ablation.

For the reported PIF benchmark, we select λ_{mass} on the validation settings (2- and 3-way), choosing the smallest value satisfying validation $p(\text{invalid}) \leq 0.01$ and conditional TV ≤ 0.02. This gives $\lambda^*_{\text{mass}}{=}0.5$. Evaluated on the locked nine-way setting over three seeds, the mass-aware objective reduces off-support probability from $0.35{\pm}0.04$ to $0.0004{\pm}0.0002$, while preserving conditional TV at 0.017.

Experiment	λ_{mass}	steps	train domains	selection	seeds
Uniform installer	0.05	300	6 train	fixed	5
Per-target biased	0.05	300	per-target	fixed	5
Universal adapter	0.05	500	6 train	fixed	3
PIF benchmark (reported)	0.5	250	per-setting	val-selected	3
$\lambda_{\text{mass}}{=}0$ ablation	0	250	per-setting	fixed	1
Mass-reward sweep	$\{0, .5, 1, 2, 4\}$	250	s2/s3 $\rightarrow$ s9b	val-selected	3
LoReFT $r{=}4$ (held-out)	0.05	300	6 train	fixed	3
LoReFT $r{=}4$ (PIF)	0.5	250	per-setting	val-selected	3
Trained rank-1 edit	none	150	4 train ($N{=}32$)	fixed	1; 10[†]

Table 17: **Mass-reward weights and training configurations.** Learning rates are 2×10^{-4} for Conditional LoRA, 8×10^{-3} for LoReFT, and 5×10^{-3} for the trained rank-1 edit. [†]Figure 25 uses one seed; the robustness check of Figure 26 uses ten.

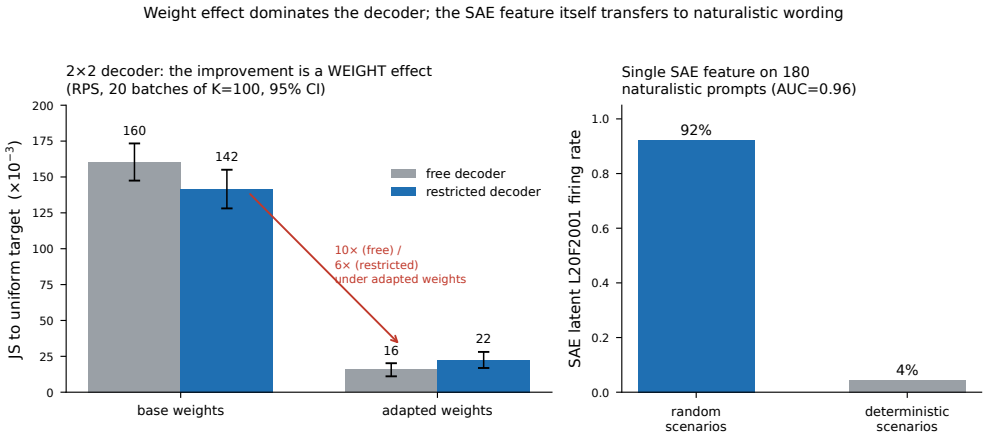

Figure 23: **Weight and SAE controls. Left:** adapting the weights improves RPS calibration under both free generation and restricted decoding ($\sim 10\times$ and $\sim 6\times$, respectively), showing that the effect is not produced by the restricted decoder. **Right:** the selected SAE feature remains selective on the 180 naturalistic prompts, firing on 92% of random-choice prompts versus 4% of controls.

D.9 Where the entropy-control effect acts: prompt vs. answer positions

The steering interventions above add the mode vector to the residual stream at *every* sequence position, yet the direction is estimated from the final-prompt-token activation. To test whether the entropy-control effect is localized to the answer-token representation or is carried by the model's processing of the prompt, we repeat the diff-in-means steering at the choice-state reader layer under three position masks: **all** positions (as above), **final token only** (the answer position under left padding), and **all except the final token** (prompt positions only). All three arms use an identical restricted-answer readout at the answer position, so absolute entropies differ slightly from the two-step marginal used in the main tables but the three-way comparison is exact. We restrict interpretation to coefficients that preserve valid-answer mass (≥ 0.2); at larger coefficients Llama's valid mass collapses and the resulting near-uniform entropy reflects breakage of the model.

The finding replicates a common qualitative conclusion with a quantitative difference between models (Fig. 24). On *both* models, injecting the vector at the answer position *alone* is insufficient to reproduce the full-sequence effect, and prompt-only injection recovers the majority of it. On Gemma-2-9B the localization is stark: final-token-only steering leaves deterministic entropy at baseline ($0.17 \rightarrow 0.17$ bits at the largest mass-preserving coefficient, $\sim 0\%$ of the all-position gain), while prompt-only injection recovers $\sim 85\%$. On Llama-3.1-8B the answer token carries a substantial share (final-token-only recovers ~ 36–62% across the mass-preserved range), but is still not sufficient alone. We therefore do *not* claim that the intervention edits an answer-position representation: the mode direction acts as a distributed, largely prompt-borne control signal that reconfigures how the answer is computed, consistent with the broadly-encoded "switch" state of §4.4.

D.10 Does Mechanism Guidance Help a Trained Rank-1 Edit?

We ask whether the mechanistic reader–writer decomposition provides a useful initialization for a learned rank-1 intervention. We train

$$h \mathrel{+}= \alpha\, v\, (\hat{u} \cdot \hat{h}),$$

with $2d$ learnable parameters on the 32 random prompts from the training domains. The objective uses the KL-to-uniform and preservation terms of §5.1, but omits the valid-mass reward. We vary only the initialization and insertion layer (Fig. 25).

Mechanistic initialization at the identified layer reaches held-out entropy 2.31 with valid mass 0.67 and deterministic $p_{\text{modal}} = 0.96$. Random initialization reaches 1.84 at mass 0.44, while the same mechanistic initialization inserted at a random late layer reaches 1.83 at mass 0.56. Both fall well

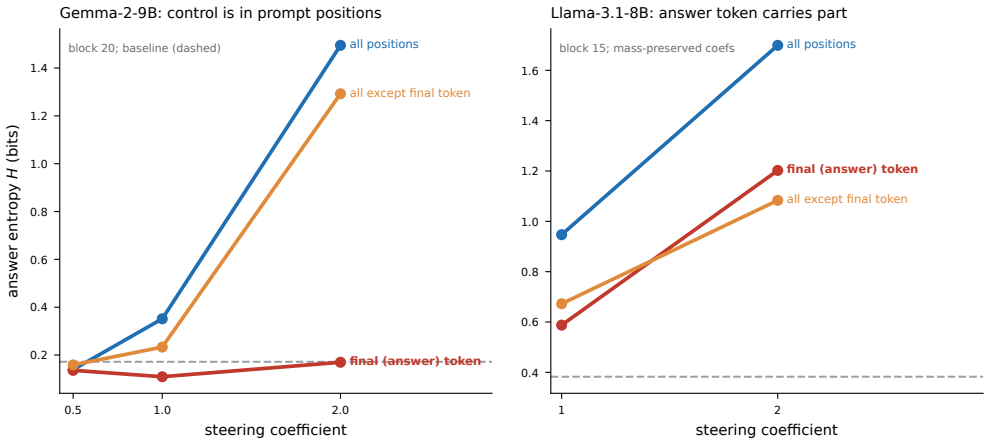

Figure 24: **The entropy-control effect spreads across prompt positions beyond the answer token.** Deterministic-prompt answer entropy under diff-in-means steering at the choice-state reader layer, for three position masks; coefficients shown preserve valid-answer mass (≥ 0.2). Final-token-only steering (red) is insufficient on both models, flat at baseline on Gemma, partial on Llama, while all-but-the-answer-token (orange) tracks the full all-position effect (blue).

short of the correctly placed mechanistic intervention. Because the objective contains no mass reward, valid mass is diagnostic rather than optimized. In particular, the entropy increase under random initialization is accompanied by lower mass and is therefore partly consistent with the off-support leakage discussed in Appendix D.2. Only mechanistic initialization at the identified layer jointly produces high entropy and high valid mass.

Thus, within this rank-1 training setup, both the mechanistic direction and its insertion depth are important: training from random initialization does not reliably recover the same edit. We use the complete 32-prompt training split here and do not study scaling with training-set size. This experiment tests whether mechanism knowledge helps a highly constrained rank-1 learner; it is not intended as a comparison over the full space of training methods.

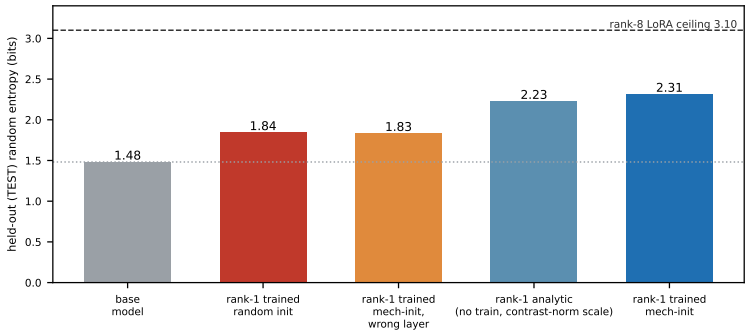

Figure 25: **Mechanistic initialization enables a trained rank-1 edit.** All trained variants use the same 32 prompts and $2d$ parameters. Mechanistic initialization raises valid mass even at the wrong layer, but only at the identified mechanism layer does it jointly achieve high entropy and high mass $(2.31, 0.67)$. Random initialization performs substantially worse. The training objective contains no valid-mass reward, so mass is shown as a diagnostic.

Robustness across seeds and depth. We next repeat training over 10 random seeds and sweep the insertion depth (Fig. 26). Across seeds, random initialization reaches entropy 1.76 ± 0.12 (range 1.51–1.91) at valid mass 0.47 ± 0.02. Mechanistic initialization reaches 2.31 ± 0.003 (range 2.31–2.32) at mass 0.67 ± 0.001. The separation is therefore much larger than the observed seed-to-seed variation on either metric.

The untrained analytic edit also shows a localized depth profile: entropy peaks at blocks 29–31 (2.24/2.23) and falls to 1.61/1.62 at blocks 23/40. The mechanistically initialized trained edit follows the same profile. By contrast, random-initialized training at blocks 26, 31, and 37 reaches only 1.51, 1.50, and 1.83, respectively. Across the tested depths, random initialization therefore does not reproduce the mechanistic edit, whereas mechanistic initialization succeeds near the same late-layer region identified by the analytic intervention.

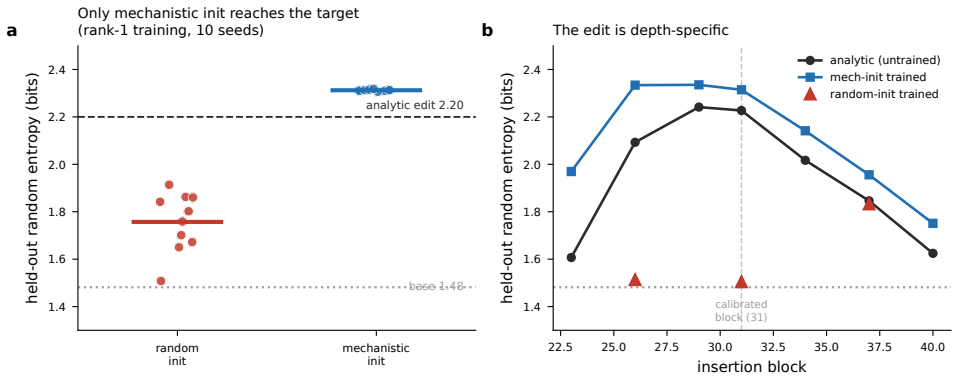

Figure 26: **Seed and depth robustness of the rank-1 edit. (a)** Held-out random entropy across 10 training seeds under random and mechanistic initialization (bars: means; dotted: base 1.48; dashed: analytic edit 2.20). **(b)** Depth sweep of the analytic untrained edit (○), mechanistically initialized trained edit (■), and random-initialized trained edit (▲). The analytic curve uses the normalization of Appendix D.4 and peaks near blocks 29–31; the trained mechanistic edit follows the same localized profile, whereas random-initialized training remains ineffective at the tested depths.

D.11 Recent-Generation Replication: Gemma-3-12B-IT

To test whether the main findings survive a model-generation change, we repeat the pipeline on `gemma-3-12b-it` using the same 208-prompt design and evaluation protocol. We evaluate the linear representation, SAE features and attribution, hand-built steering, Conditional LoRA, LoReFT, and the PIF benchmark. SAE analyses use public Gemma Scope 2 residual JumpReLU SAEs.

Behavioral premise. Across the full behavioral prompt set, Gemma-3 is the most biased sampler we measure: mean random-request entropy is 0.54 bits, compared with the mixed-domain uniform reference of 3.09 bits and deterministic entropy of 0.07 bits. Several preferred answers are inherited from Gemma-2, including digit 7, number 7, letter Q, October, and Wednesday.

Linear representation. The random-or-arbitrary choice representation replicates cleanly. Leave-one-domain-out performance and explicit-to-implicit transfer are both 1.00, with mention-FPR 0.00 from block 8 onward; the best layer is block 11. Earlier layers remain sensitive to the surface cue: mention-FPR reaches 0.75 despite probe AUC 1.00. Thus, the early lexical detector–later semantic representation dissociation also appears in Gemma-3.

SAE choice-state feature. Using the same random-versus-deterministic firing-gap criterion as in the main analysis, Gemma Scope 2 identifies `L24F5418`. The feature fires on 98% of random requests, including 100% of implicit and 96% of explicit random requests, compared with 12% of deterministic prompts, 0% of mention controls, and 0.3% of Pile tokens. A highly selective SAE feature therefore survives both the model-generation and SAE-generation change.

Attribution-selected writer. Late-stack gradient attribution identifies `L41F13390`. Adding its decoder direction raises random-request entropy from 0.54 to 1.89 at coefficient 1 while preserving valid mass at 0.48, and reaches 2.23 bits at coefficient 2. Five random-latent controls reach at most 0.92 bits at the matched coefficient, or raise entropy only by reducing valid mass. In contrast to Gemma-2, ablating the most attributed peakedness latents changes entropy by at most 0.12 bits. The clearest writable handle in Gemma-3 is therefore the additive uniformity direction.

Rank-1 reader–writer edit. Under the split discipline of §5.2, the reader is the training-domain difference-in-means direction at block 41, and α is selected on validation data before a single locked-test evaluation. The raw cosine gate used for Gemma-2 transfers poorly: late Gemma-3 residuals contain a large shared component along the reader direction, with mean test cosine 0.59 on random prompts and 0.51 on deterministic prompts. The gate therefore passes approximately 86% of full write strength even on deterministic prompts. With α constrained to preserve validation deterministic accuracy at ≥ 0.95, the edit recovers only 19% of the trained entropy gain. Relaxing the constraint to ≥ 0.90 raises recovery to 60%, but test deterministic accuracy falls to 0.72.

A simple centering correction restores conditionality. Using only the training set, we define

$$g = \mathrm{ReLU}(\cos - b), \qquad b = \mathbb{E}_{\text{train,det}}[\cos].$$

On the locked test domains, this calibrated gate raises random entropy from 0.80 to 2.06, while valid mass changes from 0.31 to 0.33. It recovers approximately 55% of the trained entropy gain while leaving deterministic accuracy unchanged (0.976 → 0.980). Gating with the SAE choice-state feature `L24F5418` gives a comparable result; its mean training activation is 378 on random prompts versus 86 on deterministic prompts. Thus, the reader–writer construction remains usable in Gemma-3, but the reader gate must be centered because deterministic residuals no longer project near zero onto the mode direction.

Conditional LoRA. On the held-out test domains, the base entropy is 0.803 bits. Across three seeds, Conditional LoRA raises it to 3.069, 3.067, and 3.060 bits. Valid mass rises from 0.313 to 0.855, 0.883, and 0.858, respectively. Deterministic p_{modal} changes from 0.986 to 0.954, 0.953, and 0.954. The two capability spot checks remain stable: tinyMMLU stays at 0.67, while tinyGSM8K changes from 0.92 to 0.91.

PIF benchmark. Gemma-3 shows the same overall pattern on the final PIF benchmark: conditional LoRA remains close to the finite-sample PRNG reference across all five settings, whereas prompting remains substantially less calibrated (Table 2).

Optimization and LoReFT. Under the same fixed rank-4 LoReFT recipe and mass-aware validation rule used for the all-model PIF benchmark, Gemma-3 is stable on all five settings. Across the five PIF settings, mean empirical JS divergence ($\times 10^{-3}$) is 1.3/1.8/2.5/2.0/10.9 for 2-choice, biased-2, 3-choice, biased-3, and biased-9, respectively (Table 2). No Gemma-3-specific retuning is used in the reported cross-model benchmark.

Limits of transfer. Two aspects of the Gemma-2 control geometry do not transfer directly. First, although the difference-in-means direction remains perfectly linearly readable, direct additive steering with it does not produce a sufficient mode switch at any of seven tested depths (blocks 12, 16, 24, 32, 36, 40, and 44). Entropy increases only as valid-answer mass collapses, similarly to a norm-matched random control. Thus, the readable representation transfers more robustly than the exact additive steering geometry.

Second, the uniform-install LoReFT operating point shows a stronger calibration–capability trade-off on Gemma-3. It reaches held-out entropy 2.73 at valid mass 0.68, but reduces tinyGSM8K from 0.92 to 0.67. A gentler operating point improves retention at the cost of weaker calibration. Conditional LoRA does not exhibit this capability loss.

D.12 Recent-Generation Replication: Qwen3-8B

We repeat the representation and control experiments on `qwen3-8b` using the same prompt design and evaluation protocol. We do not include an SAE analysis because we do not use a compatible public SAE substrate for this checkpoint.

Behavioral premise. Across the full behavioral prompt set, Qwen3 remains strongly biased on random-choice requests, with mean random-request entropy 1.16 bits. Its preferred answers differ from the Gemma family; for example, its most frequent digit response is 4 rather than 7.

Linear representation. The random-or-arbitrary choice representation again transfers cleanly. Leave-one-domain-out performance and explicit-to-implicit transfer are both 1.00, with mention-FPR 0.00 at the best mid-stack layer. The strongest probe occurs around block 9 of 36. As in the other checkpoints, earlier layers are more sensitive to explicit lexical cues, while the later representation generalizes to implicit choice-state prompts.

Additive steering. The difference-in-means direction provides only a weak causal handle in Qwen3. Steering produces small entropy changes and partial effects in both directions rather than the clean one-directional mode switch found in Gemma-2. As in Gemma-3, the linearly readable representation therefore transfers more robustly than its exact additive steering geometry.

Conditional LoRA. On the held-out test domains, base random entropy is 1.21 bits. Across three seeds, Conditional LoRA raises it to approximately 3.07 bits, while valid mass increases from 0.18 to approximately 0.80. Deterministic p_{modal} falls from 0.905 to approximately 0.797, reproducing the task-local deterministic cost observed for Qwen2.5. General-capability retention remains stable on the evaluated spot checks: tinyMMLU stays at 0.77, while tinyGSM8K changes from 0.88 to 0.89.

PIF benchmark. Conditional LoRA remains close to the finite-sample PRNG reference across the five benchmark settings. In contrast, SSoT remains substantially biased under the non-reasoning configuration despite often producing syntactically parseable outputs. As in Qwen2.5, output-format compliance alone therefore does not remove the underlying distributional bias.

LoReFT. Low-rank representation control also transfers to Qwen3. Under the uniform-install objective, LoReFT approaches the entropy ceiling while preserving tinyGSM8K performance; tinyMMLU changes only slightly ($0.77 \rightarrow 0.75$). Per-setting PIF control is also effective, although some small-support biased targets are less stable across seeds and optimization settings.

Overall, Qwen3 reproduces the domain-general choice-state representation and the effectiveness of learned conditional control, while providing further evidence that the exact additive mode-switch geometry is not invariant across models or generations.