THINKING LEAKAGE: A CAUSAL AUDIT OF NOTHINK POST-TRAINING IN HYBRID REASONING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training hybrid reasoning models in NOTHINK mode has attracted growing interest as a way to improve performance while keeping inference fast. However, these gains may draw on thinking behavior already accessible through the base model's THINK mode. We formulate this *thinking leakage* in a causal mediation framework and audit its contribution using bidirectional interventions along a simple base-derived activation direction. Across three models and three post-training methods on competition math benchmarks, we find that leakage is real, causal, and substantial: behavioral and representational analyses reveal shifts toward THINK, steering the base model along this direction reproduces most of the post-training accuracy gain, and counter-steering a checkpoint removes a substantial share of what it gains. Across nine aligned checkpoints with positive NOTHINK gains, the resulting leakage ratio ranges from 42% to 79%. These interventions support a substantial causal contribution of thinking leakage. Our findings show that a post-training method's apparent advantage can therefore reflect greater drift toward THINK, obscuring whether it improves capability within NOTHINK or more effectively re-invokes existing THINK behavior. Code is available at `https://anonymous.4open.science/r/thinking-leakage-A9F8`.

1 INTRODUCTION

Hybrid reasoning models use a single set of weights to support two inference modes (THINK and NOTHINK) for problems of varying difficulty (Yang et al., 2025; DeepSeek-AI et al., 2025; Google DeepMind, 2026; NVIDIA, 2026; KimiTeam et al., 2026). In THINK, the model generates an explicit *chain of thought* before responding to a prompt; in NOTHINK, it responds without the benefit of such a chain of thought. Proprietary models from OpenAI and Anthropic also offer finer control through multiple levels of reasoning effort (OpenAI, 2026; Anthropic, 2026). Explicit reasoning can improve performance on challenging mathematical problems (Snell et al., 2024), whereas NOTHINK reduces inference cost (Sui et al., 2025).

Recent work has explored the post-training of hybrid models in the NOTHINK mode using reinforcement learning (Zhu et al., 2025b; Lin et al., 2026; Xu et al., 2026; Huang et al., 2026), self-distillation (Hübotter et al., 2026; Li et al., 2026a;b), and on-policy distillation (Yang et al., 2026; Ding & Zhang, 2026), reporting substantial gains on reasoning benchmarks. However, this setting does not establish the source of the reported gains.

Post-training can elicit abilities that are latent in a base model (Yue et al., 2025; Karan & Du, 2026; Zhu et al., 2025a). In hybrid reasoning models, however, strong reasoning abilities are already accessible through THINK mode. Consistent with this distinction, checkpoints post-trained in NOTHINK mode still fall short of the base model's THINK performance when evaluated in NOTHINK (section 3). Thus, improved NOTHINK performance need not reflect reasoning capability newly elicited by post-training. We call this possibility *thinking leakage*: a NOTHINK post-training gain that depends on increased engagement of reasoning behavior already accessible through the base model's THINK mode.

Related behaviors have been observed in prior work (Zhu et al., 2025a; Gan et al., 2026; Zhang et al., 2025; Tu et al., 2025). Here, we ask whether *thinking leakage* occurs during NOTHINK post-training and, if so, whether it merely correlates with performance gains or causally contributes to them—and by how much.

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.

We formalize these questions with a causal model in which thinking leakage mediates the effect of post-training on NoThink accuracy, and operationalize leakage through a direction in the residual stream. Measurements of internal representations and generated behavior establish the presence of leakage, while interventions along this direction test its causal contribution. The resulting causal decomposition separates the observed gain into a component causally dependent on the audited leakage direction and a residual component, yielding the *leakage ratio*. Across three models and three post-training methods, we find that this dependence is substantial: among nine aligned checkpoints with positive NoThink gains, the leakage ratio ranges from 42% to 79%.

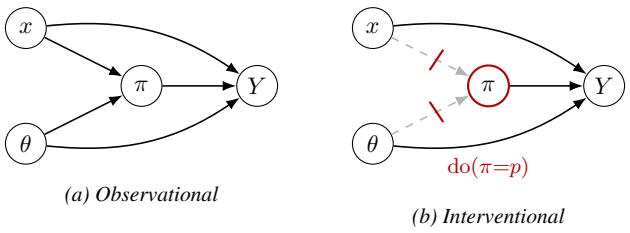
Our contributions are as follows.

- We formalize *thinking leakage* as a causal mediator of NoThink post-training gains and operationalize it as a single direction in the residual stream along which post-training systematically displaces the NoThink state toward the Think state.
- We establish the causal role of this mediator through bidirectional interventions. Moving base-model NoThink representations along the leakage direction reproduces much of the post-training gain, while reversing the displacement in post-trained models removes it, showing that a substantial share of the gain is mediated by reasoning behavior already accessible to the base model.
- We derive a *leakage ratio* that quantifies the fraction of each positive NoThink gain causally dependent on the audited leakage direction. Across the models, training methods, and checkpoints we audit, the leakage-dependent component scales approximately linearly with representational drift, with a single model-specific coefficient accounting for methods and training steps.

2 A Causal Account of Thinking Leakage

2.1 Problem Formulation

Hybrid models such as Qwen3 (Yang et al., 2025) and MiniCPM (MiniCPMTeam et al., 2025) expose Think and NoThink modes through the same parameters. We ask whether NoThink post-training gains arise partly from re-invoking computation already accessible in Think, and how much of the gain depends on that re-engagement.



(a) Observational *(b) Interventional*

Figure 1: Causal structure of a NoThink forward pass. **(a)** Observationally, thinking engagement π is generated by the input x and weights θ; accuracy Y mixes the direct path $\theta \to Y$ with the mediated path $\theta \to \pi \to Y$. **(b)** Steering and counter-steering intervene on π, changing its natural value while leaving $\theta \to Y$ and $x \to Y$ intact: outcomes under intervention still depend on the weights.

Post-training can affect accuracy Y directly and through engagement π (fig. 1). Observation alone cannot separate these paths, so we intervene on π while holding weights fixed. With θ_0, θ_1 denoting base and post-trained weights and π_0, π_1 their natural engagement levels, define

$$\begin{aligned} Y_{0,0} &= Y(\theta_0, \pi_0), \quad Y_{0,1} = Y(\theta_0, \pi_1), \\ Y_{1,0} &= Y(\theta_1, \pi_0), \quad Y_{1,1} = Y(\theta_1, \pi_1). \end{aligned} \tag{1}$$

$Y_{0,0}$ and $Y_{1,1}$ are observed; steering estimates $Y_{0,1}$ and counter-steering estimates $Y_{1,0}$, enabling the decomposition in section 6.

2.2 THE LEAKAGE AXIS

Let $\bar{h}^{(\ell)}_{\mathcal{M}}(x;\theta)$ be the residual-stream activation at layer ℓ, averaged over generated positions from mode $\mathcal{M} \in \{\mathcal{T}, \mathcal{N}\}$ and its rollouts for problem x, and $\bar{h}^{(\ell)}_{\mathcal{M}}(\theta)$ its dataset mean:

$$\bar{h}^{(\ell)}_{\mathcal{M}}(x;\theta) = \frac{1}{|\mathcal{M}(x)|} \sum_{t \in \mathcal{M}(x)} h^{(\ell)}_t(x;\theta), \quad \bar{h}^{(\ell)}_{\mathcal{M}}(\theta) = \mathbb{E}_x[\bar{h}^{(\ell)}_{\mathcal{M}}(x;\theta)]. \tag{2}$$

Following difference-of-means approaches (Marks & Tegmark, 2023; Arditi et al., 2024), we define the *leakage vector* and *leakage axis* as

$$v^{(\ell)}_{\text{leak}} = \bar{h}^{(\ell)}_{\mathcal{T}}(\theta_0) - \bar{h}^{(\ell)}_{\mathcal{N}}(\theta_0), \qquad \hat{v}^{(\ell)}_{\text{leak}} = v^{(\ell)}_{\text{leak}} / \|v^{(\ell)}_{\text{leak}}\|. \tag{3}$$

We operationalize engagement as the NOTHINK projection

$$\pi^{(\ell)}(x;\theta) = \langle \bar{h}^{(\ell)}_{\mathcal{N}}(x;\theta), \hat{v}^{(\ell)}_{\text{leak}} \rangle, \qquad \pi^{(\ell)}_{\theta} = \mathbb{E}_x[\pi^{(\ell)}(x;\theta)], \tag{4}$$

and normalize checkpoint drift by the base THINK–NOTHINK gap:

$$\Delta\pi = (\pi^{(\ell)}_1 - \pi^{(\ell)}_0) / \|v^{(\ell)}_{\text{leak}}\|. \tag{5}$$

Thus base NOTHINK is 0 and base THINK lies at 1 on the normalized displacement scale. Unless noted otherwise, we audit layer $\ell = 20$.

2.3 EXPERIMENTAL SETUP

We audit Qwen3-8B, Qwen3-4B (Yang et al., 2025), and MiniCPM4.1-8B (MiniCPMTeam et al., 2025), post-trained only in NOTHINK using GRPO (Guo et al., 2025; Shao et al., 2024), SFT, or OPSD with or without LoRA (Zhao et al., 2026). Evaluation uses 120 competition-math problems, 30 each from AIME24 (Zhang & Math-AI, 2024), AIME25 (Zhang & Math-AI, 2025), HMMT-Feb-2025, and HMMT-Nov-2025 (Dekoninck et al., 2026). Full training, sampling, intervention, and evaluation details are in section C.

3 DOES LEAKAGE OCCUR? DETECTING DRIFT IN NOTHINK

We first test whether post-training shifts NOTHINK toward THINK. At layer 20 we measure: (i) alignment $c_s = \cos(\delta_s, \hat{v}_{\text{leak}})$ of checkpoint drift $\delta_s = \bar{h}_{\mathcal{N}}(\theta_s) - \bar{h}_{\mathcal{N}}(\theta_0)$; (ii) problem-level paired projection drift summarized by repeated-measures Cohen's d; (iii) density of THINK-characteristic reasoning markers; and (iv) the checkpoint-specific THINK–NOTHINK mode gap. Formally,

$$c_s = \frac{\pi_s - \pi_0}{\|\delta_s\|}, \qquad d_s = \frac{\overline{\Delta}}{s_\Delta}, \quad \Delta_q = \pi(x_q;\theta_s) - \pi(x_q;\theta_0), \tag{6}$$

and marker density is

$$b(x) = \frac{\#\{\text{reasoning markers in } x\}}{\#\{\text{generated tokens in } x\}}. \tag{7}$$

For geometry we recompute the two mode centroids within each checkpoint:

$$v_{\text{leak}}(\theta) = \bar{h}_{\mathcal{T}}(\theta) - \bar{h}_{\mathcal{N}}(\theta), \qquad \phi(\theta) = \angle\big(\bar{h}_{\mathcal{T}}(\theta), \bar{h}_{\mathcal{N}}(\theta)\big). \tag{8}$$

Layerwise analyses are in sections D.2 and D.3.

Leakage occurs, and it scales with the reported gain. Figure 2 plots the first three measures against NOTHINK accuracy across all checkpoints. All three views—population-level, problem-level, and behavioral—agree: as NOTHINK accuracy improves, representations drift toward the THINK state and generated responses become more THINK-like ($\rho = +0.92$, $+0.68$, and $+0.85$, respectively). The highest-accuracy checkpoints exhibit the strongest THINK-ward drift, while the few checkpoints that lose accuracy relative to base move in the opposite direction along the leakage axis. Thus, across checkpoints, improvements in NOTHINK accuracy are tightly coupled to increased engagement of the THINK-associated direction.

The coupling is not, however, fixed across training methods. At matched NOTHINK accuracy on the Qwen models, SFT moves more than twice as far along $\hat{v}_{\text{leak}}$ as GRPO (section D.4). Accuracy alone therefore does not determine the amount of leakage, motivating the causal interventions that follow.

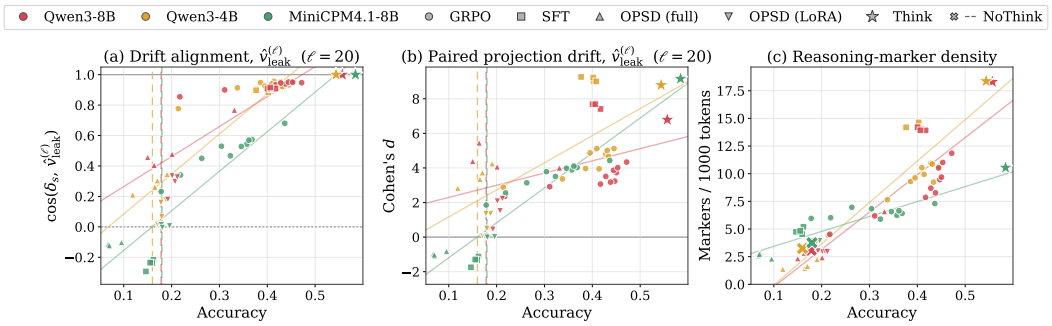


Figure 2: **Thinking leakage tracks NoThink accuracy across checkpoints of different training methods and model scales.** Each point is one checkpoint; marker shape denotes the training method and color the base model. $\times$ and $\star$ are the base model in NoThink and Think. **(a)** Drift alignment to the leakage axis $\hat{v}_{\text{leak}}$. **(b)** Paired projection drift (Cohen's d) along $\hat{v}_{\text{leak}}$. **(c)** Reasoning-marker density in the NoThink rollouts. More experiments and analysis are given in section D

The drift is specific to the reasoning-associated direction. A generic consequence of post-training—for example, learning to produce better direct answers—could also shift internal representations. We therefore repeat the representational analyses using a reference axis defined by the clean solution that a Think rollout produces *after* deliberation, rather than by the deliberative portion itself (section D.1). The contrast is sharp: the effect size along the leakage axis is more than an order of magnitude larger ($d = 3.15$ versus 0.02), and only drift along the leakage axis strongly tracks accuracy ($\rho = +0.68$ versus $+0.23$). This separation persists at every probed layer (section D.2). The observed drift therefore aligns specifically with the representation associated with deliberation, rather than with a generic shift toward better final-answer representations.

NoThink converges toward Think. The geometry of the two modes provides a complementary view. Recomputing their centroids within each checkpoint shows that GRPO reduces the Think–NoThink gap by more than half on Qwen3-8B (fig. 3). The norms of both $\bar{h}_{\mathcal{T}}$ and $\bar{h}_{\mathcal{N}}$ remain within a few percent of their base values, while 84% of the reduction in the gap is attributable to the narrowing angle between them.

This convergence is strongly asymmetric. Measured relative to the base mode gap, $\bar{h}_{\mathcal{T}}$ moves only 5.8% and 10.2% at the two checkpoints, whereas $\bar{h}_{\mathcal{N}}$ moves 53.2% and 66.2%. Thus, the shrinking separation is driven primarily by NoThink moving toward a comparatively stable Think representation, rather than by both modes moving toward a new common state. The same ordering holds at every probed layer (section D.3).

These results establish that leakage accompanies successful post-training, but not that the displacement causes the gain. We test that claim next by intervening directly on the leakage direction.

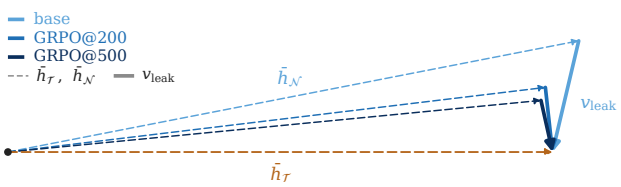


Figure 3: **GRPO collapses the mode gap.** Layer-20 geometry of Qwen3-8B at GRPO steps 0, 200, 500. $\bar{h}_{\mathcal{T}}$, $\bar{h}_{\mathcal{N}}$ (dashed), and $v_{\text{leak}} = \bar{h}_{\mathcal{T}} - \bar{h}_{\mathcal{N}}$ (solid) are recomputed within each checkpoint. Both $\bar{h}_{\mathcal{T}}$, $\bar{h}_{\mathcal{N}}$ keep their norms to within a few percent, yet the angle between them closes from $11.31°$ to $5.48°$ and $\|v_{\text{leak}}\|$ falls $15.43 \to 8.81 \to 7.18$. The panel is an approximate embedding rather than an orthogonal projection of 4096-d quantities; embedding errors are reported in section D.3.

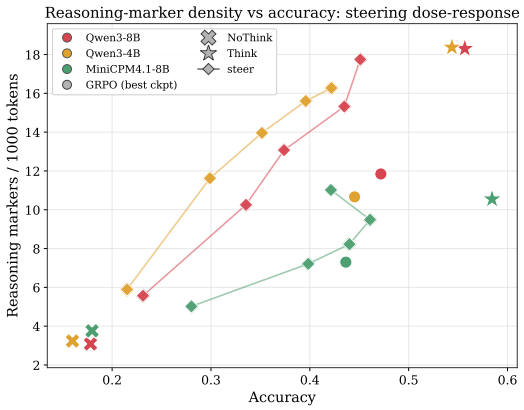


Figure 4: **Steering along $\hat{v}_{\text{leak}}$ recovers the THINK performance without training.** Reasoning-marker density versus NOTHINK accuracy, coloured by base model. Diamonds trace the layer-20 induction sweep over five doses; $\times$ and $\star$ are the unsteered base model in NOTHINK and THINK, and the circle is each model's best post-training checkpoint.

4 IS THE DIRECTION SUFFICIENT? INDUCING LEAKAGE BY STEERING

Section 3 shows that NOTHINK post-training moves representations along $\hat{v}_{\text{leak}}$, but correlation alone cannot establish whether this displacement contributes to the accompanying behavior. We therefore test its causal *sufficiency*: if we move the untouched base model's NOTHINK state toward THINK along $\hat{v}_{\text{leak}}$, without changing its weights, can we reproduce the behavioral effects associated with post-training? In the causal model, this amounts to estimating

$$Y\big(\theta_0, \mathrm{do}(\pi{=}p)\big)$$

over a range of engagement levels p and comparing it with the untreated baseline $Y(\theta_0, \pi{=}\pi_0)$.

We implement this intervention using additive activation steering (Rimsky et al., 2024). During NOTHINK generation, we add a scaled copy of the leakage vector at every generated token position at layer 20—the layer at which π is measured and a depth at which representations have been found to be causally manipulable by activation interventions (Zou et al., 2025; Rimsky et al., 2024; Cao et al., 2024):

$$h_t \;\leftarrow\; h_t + \alpha v_{\text{leak}} \qquad \Longrightarrow \qquad \pi \;\mapsto\; \pi + \alpha\|v_{\text{leak}}\| \;=:\; p_\alpha, \qquad \alpha > 0. \tag{9}$$

Because v_{leak} spans the base model's mean NOTHINK–THINK separation, α is measured in units of that gap: $\alpha = 1$ adds one full base-mode displacement along the leakage axis.

We sweep $\alpha \in \{0.5, 1, 1.25, 1.5, 2\}$ and measure how accuracy and reasoning-marker density change as the base NOTHINK state is displaced toward THINK. If steering alone moves both outcomes toward those of the post-trained checkpoints, then displacement along the leakage direction is causally sufficient to induce the corresponding behavioral changes. This forward intervention is paired with the counter-steering experiment in section 5, which reverses the same displacement to test whether it is also necessary for the observed post-training gain.

One direction reproduces most of the post-training gain. Figure 4 plots the five-dose sweep in the accuracy–marker-density plane. As α increases, the base model moves smoothly up and to the right from its NOTHINK baseline toward its own THINK point. Without any weight update, the best steered point recovers 92–110% of the best checkpoint's accuracy gain over base NOTHINK, and roughly 70% of the full NOTHINK→THINK accuracy gap on all three models. On MiniCPM4.1-8B, steering surpasses the best checkpoint; on both Qwen models, accuracy is still increasing at the largest dose tested.

This effect is specific to the leakage direction: applying the same intervention along a reference axis yields no comparable gain (section E.1), while steering along $\hat{v}_{\text{leak}}$ at other depths produces

the same qualitative trend (section E.2). Thus, displacement along a single base-model direction is causally sufficient to reproduce most of the observed post-training accuracy gain.

The axis is not exhausted by post-training. Applying the same intervention to an already post-trained checkpoint yields further improvement: steering Qwen3-8B's best GRPO checkpoint with $\alpha = 1$ raises accuracy from 0.472 to 0.553, within 0.4 percentage points of base THINK performance (section E.4).

Steering induces more reasoning markers than training. Steering and post-training do not, however, produce identical behavior. At its best dose, the steered model reaches 89–97% of the base THINK marker density, whereas the best post-trained checkpoint reaches only 58–69%. Per point of accuracy gained, steering produces 1.5–$1.9\times$ as many reasoning markers as post-training. Thus, direct activation injection drives the surface signature associated with THINK more strongly than training does, while converting that shift into accuracy less efficiently. The intervention therefore reproduces most of the accuracy gain without simply reproducing the checkpoint's entire behavioral profile.

At the largest dose, token budget becomes limiting. MiniCPM4.1-8B is the only model whose accuracy declines at $\alpha = 2$, falling 4.0 percentage points from its peak at $\alpha = 1.5$ even as reasoning-marker density continues to rise. At this dose, steering also lengthens responses sufficiently that they more often reach the token limit (section E.3). The decline is therefore consistent with a generation-budget constraint rather than a reversal of the steering effect, and does not affect the sufficiency result at lower doses.

Sufficiency alone, however, does not show that post-trained checkpoints actually depend on the displacement they acquire. Counter-steering tests that complementary claim.

5 Does the Gain Depend on Leakage? Removing It by Counter-Steering

Section 4 shows that displacement along the leakage direction is sufficient to reproduce most of the post-training accuracy gain. We now ask the complementary question: how much of the observed gain depends on the displacement that post-training actually induces? We *counter-steer* a post-trained checkpoint θ_1, moving its NOTHINK state back along v_{leak} toward base engagement and measuring how much of the gain remains. In the causal model, this estimates

$$Y\big(\theta_1, \mathrm{do}(\pi{=}p)\big)$$

at engagement levels $p < \pi_1$, relative to the untreated outcome $Y(\theta_1, \pi{=}\pi_1)$.

By eq. (5), $\Delta\pi$ is the checkpoint's normalized engagement drift from base NOTHINK. We reverse the steering intervention of eq. (9), scaling it by the drift measured for each checkpoint. During NOTHINK generation, at layer 20 we apply

$$h_t \;\leftarrow\; h_t - \gamma\,\Delta\pi\,v_{\text{leak}} \qquad \Longrightarrow \qquad \pi \;\mapsto\; \pi - \gamma\,\Delta\pi\,\|v_{\text{leak}}\| \;=:\; p_\gamma, \qquad \gamma > 0. \tag{10}$$

The dose is therefore expressed in units of the checkpoint's own drift. Our primary intervention uses $\gamma = 1$, which exactly removes the checkpoint's acquired displacement along the leakage axis and, by construction, returns its mean projected engagement π to the base NOTHINK level. Any resulting loss in accuracy measures the extent to which the checkpoint's gain causally depends on that displacement; the corresponding change in reasoning-marker density provides a behavioral check. This reversal forms the basis of the gain decomposition in section 6.

Checkpoint selection. For each model, we counter-steer the best checkpoint from each post-training method (*GRPO/SFT/OPSD best*), an intermediate *GRPO@200* checkpoint, and the most severely collapsed OPSD checkpoint, included to test the intervention under reverse drift as well as successful training. Based on performance and drift alignment (section 3), the fifteen checkpoints fall into three groups (section F.1). Here we focus on the nine *aligned* checkpoints, which improve over base while drifting toward THINK, and the two *reverse-drift* checkpoints, which lose accuracy while moving in the opposite direction. The remaining checkpoints and complete grouping criteria are reported in section F.1.

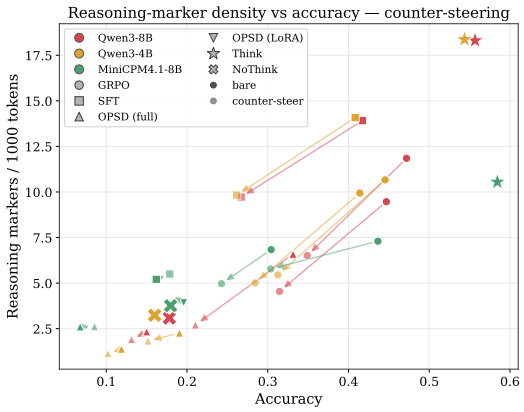


Figure 5: **Counter-steering along $\hat{v}_{\text{leak}}$ removes the post-training gains.** Reasoning-marker density versus NoThink accuracy, coloured by base model. Each checkpoint is a before $\rightarrow$ after pair joined by an arrow; $\times$ and $\star$ are the unsteered base model in NoThink and Think.

Counter-steering removes much of the post-training gain. Across the nine aligned checkpoints, counter-steering reverses a substantial fraction of both the accuracy and reasoning-marker gains acquired during post-training (fig. 5). Relative to each checkpoint's improvement over base NoThink, counter-steering removes, on average, 54.5% of the accuracy gain and 63.8% of the marker-density gain. The largest reversal occurs for Qwen3-8B OPSD@25, where 78.9% of the accuracy gain is removed. Per-checkpoint results are reported in table 8.

The effect follows the sign of the learned drift. The two reverse-drift checkpoints provide a signed control. Because their post-training displacement along v_{leak} is negative relative to base NoThink, the same $\gamma = 1$ intervention adds v_{leak} rather than subtracting it. Their accuracy *increases*. Thus, across both aligned and reverse-drift checkpoints, counter-steering consistently reverses the displacement induced by post-training: it removes accuracy when training moves the model toward Think, and restores accuracy when training moves it away. This sign-sensitive reversal is difficult to explain as generic intervention damage, which would instead predict degradation in both cases. Consistent with this interpretation, matched interventions along a random direction or the reference axis v_{ref} leave accuracy and reasoning-marker density essentially unchanged (section F.2).

Counter-steering moves behavior back toward base. Counter-steering also increases degenerate repetition, raising an alternative explanation: perhaps the intervention lowers accuracy simply by damaging generation rather than by removing leakage. Two checks argue against this account. First, at the problem level, the induced repetition tracks the base model's own repetition pattern, while the checkpoint's accuracy pattern shifts toward that of the base model as the intervention dose increases. Second, restricting the analysis to rollouts that terminate normally preserves 91% of the observed accuracy loss. Thus, only a small fraction of the reversal can be attributed to degenerate generation; most persists among normally terminating responses (section F.7).

6 How Much of the Gain Is Thinking Leakage?

The preceding experiments establish both directions of the causal test: moving the base model forward along the leakage axis improves performance, while removing acquired displacement from post-trained checkpoints reduces it. We now combine these interventions to quantify how much of each observed gain depends on that displacement.

Of the four potential outcomes in eq. (1), $Y_{0,0}$ and $Y_{1,1}$ are observed directly. Counter-steering yields $Y_{1,0}$; steering the base by the same normalized displacement yields $Y_{0,1}$, interpolated at $\alpha^\star = \Delta\pi$. The observed gain has the exact decomposition

$$\Delta = Y_{1,1} - Y_{0,0} = \underbrace{(Y_{1,1} - Y_{1,0})}_{L_{\text{trained}}} + \underbrace{(Y_{1,0} - Y_{0,0})}_{R_0}. \tag{11}$$

$$\Delta = \underbrace{Y_{1,1} - Y_{1,0}}_{L_{\text{trained}}} + \underbrace{Y_{1,0} - Y_{0,0}}_{R_0} = \underbrace{Y_{0,1} - Y_{0,0}}_{L_{\text{base}}} + \underbrace{Y_{1,1} - Y_{0,1}}_{R_1}. \tag{12}$$

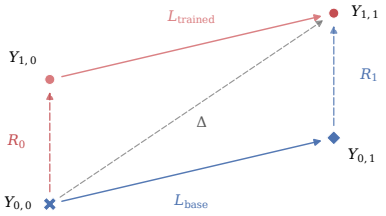


Figure 6: **The two decompositions of the total gain.** Left: the algebraic decomposition of Δ into two paths, $(L_{\text{trained}}, R_0)$ and (L_{base}, R_1). Right: a schematic illustration of the same decomposition.

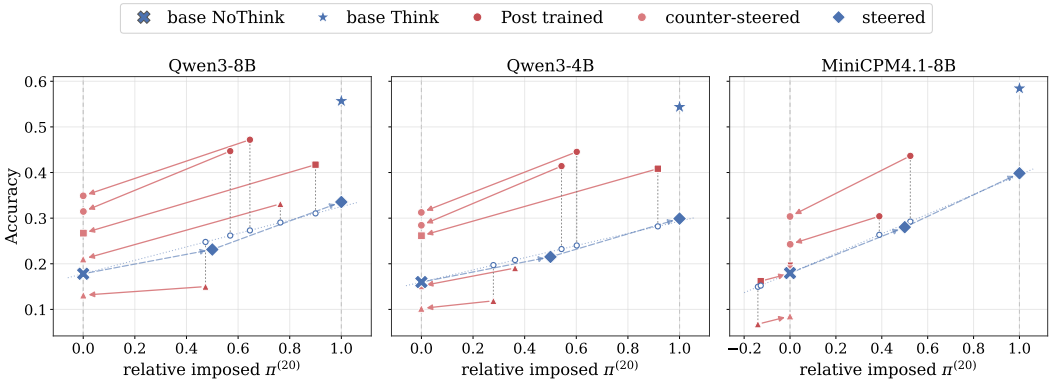


Figure 7: **NoThink accuracy against the engagement**, at layer 20, one panel per model. Blue: the base model under steering, tracing $Y_{0,0} \to Y_{0,1}$. Red: each post-trained checkpoint and its counter-steered counterpart, tracing $Y_{1,1} \to Y_{1,0}$. Arrows show the direction of the intervention, not of the decomposition. $L_{\text{trained}} = Y_{1,1} - Y_{1,0}$ is the height the checkpoint loses along the red arrow, and $R_0 = Y_{1,0} - Y_{0,0}$ compares the counter-steered checkpoint with the untreated base.

L_{trained} is the accuracy lost when acquired leakage displacement is removed; R_0 is what remains after projected engagement is restored to base.

The complementary path through $Y_{0,1}$ (the second path in fig. 6) asks the same question from the untouched base model: how much accuracy is induced when the base is moved forward by exactly the displacement acquired during training. Agreement between these two paths therefore provides a check that the estimated leakage contribution is not peculiar to either steering the base or counter-steering the trained checkpoint.

Leakage scales linearly with drift. Across all 15 checkpoints, L_{trained} is approximately linear in $\Delta\pi$ within each model ($R^2 = 0.92$–0.95, fig. 7):

$$L_{\text{trained}} \approx \beta_{\text{trained}} \Delta\pi, \qquad \Delta \approx \beta_{\text{trained}} \Delta\pi + R_0, \tag{13}$$

with $\beta_{\text{trained}} = 0.166, 0.178, 0.212$ for Qwen3-8B, Qwen3-4B, and MiniCPM4.1-8B. Thus one model-specific coefficient predicts the leakage-dependent component across methods and training steps. In contrast, R_0's association with drift falls from $\rho = +0.69$ to -0.20 after controlling for total gain, while its association with total gain remains $\rho = +0.97$ after controlling for drift. A method-level comparison using this decomposition is deferred to section G.3.

The opposite intervention agrees. The complementary path gives

$$\Delta = L_{\text{base}} + R_1, \qquad L_{\text{base}} \approx \beta_{\text{base}} \Delta\pi, \tag{14}$$

with $\beta_{\text{base}} = 0.147, 0.133, 0.215$. The ratios $\beta_{\text{trained}}/\beta_{\text{base}}$ are $1.13, 1.34, 0.99$, and the checkpoint-level asymmetry

$$\mathcal{A} = L_{\text{trained}} - L_{\text{base}} = R_1 - R_0 \tag{15}$$

averages only 2.6 accuracy points in absolute value. Base steering and checkpoint counter-steering therefore assign similar accuracy changes to the same displacement; problem-level agreement is in section G.2.

Leakage ratio. For positive-gain checkpoints, define

$$\lambda = \frac{L_{\text{trained}}}{\Delta} = \frac{Y_{1,1} - Y_{1,0}}{Y_{1,1} - Y_{0,0}}. \tag{16}$$

Using $L_{\text{trained}} \approx \beta_{\text{trained}} \Delta\pi$, this is equivalently $\lambda \approx \beta_{\text{trained}} \Delta\pi / \Delta$. Across the nine aligned checkpoints, $\lambda = 0.42$–0.79: roughly 42%–79% of observed improvement depends on the acquired displacement. GRPO lies near 0.5, SFT near 0.6, and Qwen3-8B OPSD@25 reaches 0.79. The additive decomposition also applies to reverse-drift and degrading checkpoints, but for those we retain signed components rather than interpret λ as a share of positive gain (section G.1).

7 Related Work

Steering model behavior. Activation directions have been used to control traits including sycophancy, honesty, refusal, and hallucination (Rimsky et al., 2024; Zou et al., 2025; Cao et al., 2024; Chen et al., 2025; Wang et al., 2025), and reasoning behaviors such as thinking speed and backtracking (Lin et al., 2025; Venhoff et al., 2025; Sinii et al., 2025). Unlike work requiring targeted extraction, our native THINK–NOTHINK contrast yields a simple difference-of-means direction used to audit post-training rather than merely control behavior.

Causal interventions in model interpretability. Internal interventions have been used to study mediation of gender bias (Vig et al., 2020), factual recall (Meng et al., 2022), and task-specific computations (Wang et al., 2023); their interpretation depends on intervention design and evaluation (Zhang & Nanda, 2024). We use interventions in both directions to quantify how much NOTHINK gain depends on a targeted reasoning-associated direction. Extended related work is in section A.

8 Conclusion

We investigate whether reported NOTHINK post-training gains depend on *thinking leakage*: increased engagement of reasoning behavior already accessible through the base model's THINK mode. Across three hybrid reasoning models and three post-training methods, NOTHINK improvements are accompanied by drift toward THINK in representation and behavior. A base-derived direction both reproduces most of the gain when injected and removes a substantial fraction when reversed. Among nine positive-gain aligned checkpoints, 42%–79% of improvement disappears when projected engagement is restored to base NOTHINK, and the amount removed scales approximately linearly with drift across methods and training steps.

Thus disabling explicit thinking at the interface does not ensure that post-training preserves the intended mode separation: apparent NOTHINK gains can partly recruit reasoning behavior already accessible through THINK. Auditing such methods therefore requires asking not only how much accuracy improves, but what the improvement depends on (section B.1).

9 Limitations and Future Work

Our experiments cover three models, three post-training methods, and competition mathematics; leakage may differ across architectures, scales, training procedures, and tasks. Broader coverage would clarify the generality of our findings. Moreover, a single base-derived direction need not capture every form of leakage or remain complete after large model changes. Accordingly, λ measures dependence on the audited component, not all possible leakage, and the residual R_0 cannot be identified as wholly leakage-independent or newly acquired NOTHINK capability. Multidimensional or nonlinear interventions could improve coverage and specificity (section B.2).

AI USE STATEMENT

The authors conceptualized and executed the study. Generative AI was not used to generate synthetic datasets, perform the experimental evaluations or statistical analyses, formulate mathematical claims, or assist with proofs. It was used to critique aspects of the methodology, experimental setup, and interpretation; assist with coding and figures; summarize, analyze, brainstorm, and retrieve relevant literature; and edit or rephrase text for clarity. All AI-assisted work and code were reviewed by the authors, and all citations were manually verified and added by the authors. The authors accept full responsibility for the manuscript and artifacts.

REPRODUCIBILITY STATEMENT

All models and benchmarks are public. Post-training and interventions use public frameworks; full details are in section C. Code will be released upon publication.

REFERENCES

Anthropic. Document of reasoning effort. https://platform.claude.com/docs/en/build-with-claude/effort, 2026. Claude Platform Docs. Accessed: 2026-09-11.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. 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. 49519–49551. Curran Associates, Inc., 2024. doi: 10.52202/079017-1567. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/58cbe393b4254da8966780a40d023c0b-Paper-Conference.pdf.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models, 2025. URL https://arxiv.org/abs/2507.21509.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Erhang Li, Fangqi Zhou, Fangyun Lin, Fucong Dai, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoran Wei, Haowei Zhang, Haowen Luo, Haozhe Ji, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, Jialiang Huang, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jingchang Chen, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexin Huang, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Liang Zhao, Liangsheng Yin, Lihua Guo, Lingxiao Luo, Linwang Ma, Litong Wang, Liyue Zhang, M. S. Di, M. Y Xu, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Panpan Huang, Peixin Cong, Peiyi Wang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, S. H. Liu, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Songyang Zhou, Tao Ni, Tao Yun, Tian Pei, Tian Ye, Tianyuan Yue, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjun Gao, Wentao Zhang, Xi Gao, Xiangwen Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyuan Li, Xu Chen, Xuecheng Su, Xuehai Pan, Xuheng Lin, Xuwei Fu, Y. Q. Wang, Yang Zhang, Yanhong Xu, Yanru Ma, Yao Li, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yiliang Xiong, Ying He, Ying Zhou, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu,

Yu Wu, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuduan Wang, Yue Gong, Yuhan Wu, Yuheng Zou, Yukun Li, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehua Zhao, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhiyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Zizheng Pan, Zongqing Yao, Bei Feng, Hui Li, J. L. Cai, Jiaqi Ni, Lei Xu, Meng Li, Ning Tian, R. J. Chen, R. L. Jin, S. S. Li, Shuang Zhou, Tianyu Sun, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xinnan Song, Xinyi Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, Dongjie Ji, Jian Liang, Jianzhong Guo, Jin Chen, Leyi Xia, Miaojun Wang, Mingming Li, Peng Zhang, Ruyi Chen, Shangmian Sun, Shaoqing Wu, Shengfeng Ye, T. Wang, W. L. Xiao, Wei An, Xianzu Wang, Xiaowen Sun, Xiaoxiang Wang, Ying Tang, Yukun Zha, Zekai Zhang, Zhe Ju, Zhen Zhang, and Zihua Qu. DeepSeek-V3.2: Pushing the frontier of open large language models, 2025. URL https://arxiv.org/abs/2512.02556.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with LLMs. In *3rd AI for Math Workshop: Toward Self-Evolving Scientific Agents*, 2026. URL https://openreview.net/forum?id=DmPE4byHuN.

Yi Ding and Ruqi Zhang. Does on-policy distillation really distill? from noisy teacher to self-improvement, 2026. URL https://arxiv.org/abs/2608.31046.

Siyuan Gan, Jiaheng Liu, Boyan Wang, Tianpei Yang, Runqing Miao, Yuyao Zhang, Fanyu Meng, Junlan Feng, Linjian Meng, Jing Huo, and Yang Gao. Thinking-based non-thinking: Solving the reward hacking problem in training hybrid reasoning models via reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45754–45771, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2122. URL https://aclanthology.org/2026.acl-long.2122/.

Google DeepMind. Gemma 4 model card. Google AI for Developers, 2026. URL https://ai.google.dev/gemma/docs/core/model_card_4.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao,

Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Lei Huang, Xiang Cheng, Chenxiao Zhao, Guobin Shen, Junjie Yang, Xiaocheng Feng, Yuxuan Gu, Xing Yu, and Bing Qin. Bootstrapping exploration with group-level natural language feedback in reinforcement learning, 2026. URL https://arxiv.org/abs/2603.04597.

Jonas Hübotter, Frederike Lübeck, Lejs Deen Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QkfkxyRizZ.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Vsgq2ldr4K.

KimiTeam, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jiahao Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, Yanru Chen, Yanxu Chen, Yicun Chen, Yimin Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Ziwei Chen, Dazhi Cheng, Minghan Chu, Jialei Cui, Jiaqi Deng, Muxi Diao, Hao Ding, Mengfan Dong, Mengnan Dong, Yuxin Dong, Yuhao Dong, Angang Du, Chenzhuang Du, Dikang Du, Lingxiao Du, Yulun Du, Yu Fan, Shengjun Fang, Qiulin Feng, Yichen Feng, Garimugai Fu, Kelin Fu, Hongcheng Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Chengyang Gong, Xiaochen Gong, Zhuoma Gongque, Qizheng Gu, Xinran Gu, Yicheng Gu, Longyu Guan, Yuanying Guo, Xiaoru Hao, Weiran He, Wenyang He, Yunjia He, Chao Hong, Hao Hu, Jiaxi Hu, Yangyang Hu, Zhenxing Hu, Ke Huang, Ruiyuan Huang, Weixiao Huang, Zhiqi Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yu Jing, Guokun Lai, Aidi Li, C. Li, Cheng Li, Fang Li, Guanghe Li, Guanyu Li, Haitao Li, Haoyang Li, Jia Li, Jingwei Li, Junxiong Li, Lincan Li, Mo Li, Weihong Li, Wentao Li, Xinhang Li, Xinhao Li, Yang Li, Yanhao Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zheming Li, Weilong Liao, Jiawei Lin, Xiaohan Lin, Zhishan Lin, Zichao Lin, Cheng Liu, Chenyu Liu, Hongzhang Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Tianyu Liu, Weizhou Liu, Xiangyan Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yuanxin Liu, Yue Liu, Zhengying Liu, Zhongnuo Liu, Enzhe Lu, Haoyu Lu, Zhiyuan Lu, Junyu Luo, Tongxu Luo, Yashuo Luo, Long Ma, Yingwei Ma, Shaoguang Mao, Yuan Mei, Xin Men, Fanqing Meng, Zhiyong Meng, Yibo Miao, Minqing Ni, Kun Ouyang, Siyuan Pan, Bo Pang, Yuchao Qian, Ruoyu Qin, Zeyu Qin, Jiezhong Qiu, Bowen Qu, Zeyu Shang, Youbo Shao, Tianxiao Shen, Zhennan Shen, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Feifan Song, Pengwei Song, Tianhui Song, Xiaoxi Song, Hongjin Su, Jianlin Su, Zhaochen Su, Lin Sui, Jinsong Sun, Junyao Sun, Tongyu Sun, Flood Sung, Yunpeng Tai, Chuning Tang, Heyi Tang, Xiaojuan Tang, Zhengyang Tang, Jiawen Tao, Shiyuan Teng, Chaoran Tian, Pengfei Tian, Ao Wang, Bowen Wang, Chensi Wang, Chuang Wang, Congcong Wang, Dingkun Wang, Dinglu Wang, Dongliang Wang, Feng Wang, Hailong Wang, Haiming Wang, Hengzhi Wang, Huaqing Wang, Hui Wang, Jiahao Wang, Jinhong Wang, Jiuzheng Wang, Kaixin Wang, Linian Wang, Qibin Wang, Shengjie Wang, Shuyi Wang, Si Wang, Wei Wang, Xiaochen Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yipu Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhexu Wang, Zihan Wang, Zizhe Wang, Chu Wei, Ming Wei, Chuan Wen, Zichen Wen, Chengjie Wu, Haoning Wu, Junyan Wu, Rucong Wu, Wenhao Wu, Yuefeng Wu, Yuhao Wu, Yuxin Wu, Zijian Wu,

Chenjun Xiao, Jin Xie, Xiaotong Xie, Yuchong Xie, Yifei Xin, Bowei Xing, Boyu Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinbo Xu, Xinran Xu, Yangchuan Xu, Yichang Xu, Yuemeng Xu, Zelai Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Guangyao Yang, Hao Yang, Junwei Yang, Kai Yang, Ningyuan Yang, Ruihan Yang, Xiaofei Yang, Xinlong Yang, Ying Yang, Yi Yang, Yi Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Dan Ye, Wenjie Ye, Zhuorui Ye, Bohong Yin, Chengzhen Yu, Longhui Yu, Tao Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Xiaokun Yuan, Yang Yue, Weihao Zeng, Dunyuan Zha, Haobing Zhan, Dehao Zhang, Hao Zhang, Jin Zhang, Puqi Zhang, Qiao Zhang, Rui Zhang, Xiaobin Zhang, Y. Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yushun Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Chenguang Zhao, Feifan Zhao, Jinxiang Zhao, Shuai Zhao, Xiangyu Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Junfeng Zhong, Longguang Zhong, Weiming Zhong, M. Zhou, Runjie Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yuxuan Zhu, Zhen Zhu, Jingze Zhuang, Weiyu Zhuang, Ying Zou, and Xinxing Zu. Kimi k2.5: Visual agentic intelligence, 2026. URL https://arxiv.org/abs/2602.02276.

Hynek Kydlíček. Math-Verify: Math Verification Library, 2025.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. In *Third Conference on Language Modeling*, 2026a. URL https://openreview.net/forum?id=P2OuWwZspP.

Yijiang Li, Bingyang Wang, Yijun Liang, Yunjie Tian, Di Fu, and Nuno Vasconcelos. On-policy self-distillation without any supervision, 2026b. URL https://arxiv.org/abs/2608.06296.

Zhengkai Lin, Zhihang Fu, Ze Chen, Chao Chen, Liang Xie, Wenxiao Wang, Deng Cai, Zheng Wang, and Jieping Ye. Controlling thinking speed in reasoning models. 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, Main Conference, pp. 78300–78347. Curran Associates, Inc., 2025. doi: 10.52202/085713-2623. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/70d46c12643e8f056a693ed43472dc35-Paper-Conference.pdf.

Zihan Lin, Xiaohan Wang, Jie Cao, Jiajun Chai, Li Wang, Xiaodong Lu, Wei Lin, Ran He, and Guojun Yin. ResRL: Boosting LLM reasoning via negative sample projection residual reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=kmN9ozKtGh.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

MiniCPMTeam, Chaojun Xiao, Yuxuan Li, Xu Han, Yuzhuo Bai, Jie Cai, Haotian Chen, Wentong Chen, Xin Cong, Ganqu Cui, Ning Ding, Shengda Fan, Yewei Fang, Zixuan Fu, Wenyu Guan, Yitong Guan, Junshao Guo, Yufeng Han, Bingxiang He, Yuxiang Huang, Baoxi Ji, Cunliang Kong, Qiuzuo Li, Siyuan Li, Wenhao Li, Xin Li, Yanghao Li, Yishan Li, Zhen Li, Dan Liu, Biyuan Lin, Yankai Lin, Xiang Long, Quanyu Lu, Yaxi Lu, Peiyan Luo, Hongya Lyu, Litu Ou, Yinxu Pan, Lushi Pu, Zekai Qu, Qundong Shi, Zijun Song, Jiayuan Su, Zhou Su, Ao Sun, Xianghui Sun, Peijun Tang, Fangzheng Wang, Feng Wang, Shuo Wang, Yudong Wang, Zheng Wang, Yesai Wu, Zhenyu Xiao, Jie Xie, Zihao Xie, Xiaoyue Xu, Yukun Yan, Jiarui Yuan, Jinqian Zhang, Kaihuo Zhang, Lei Zhang, Linyue Zhang, Xueren Zhang, Yudi Zhang, Hengyu Zhao, Weilin Zhao, Weilun Zhao, Yuanqian Zhao, Zhi Zheng, Chuyue Zhou, Ge Zhou, Jie Zhou, Wei Zhou, Yanghao Zhou, Zihan Zhou, Zixuan Zhou, Zhiyuan Liu, Guoyang Zeng, Chao Jia, Dahai Li, and Maosong Sun. Minicpm4: Ultra-efficient llms on end devices. *arXiv preprint arXiv:2506.07900*, 2025.

NVIDIA. Nemotron 3 nano omni: Efficient and open multimodal intelligence, 2026. URL https://arxiv.org/abs/2604.24954.

OpenAI. Document of reasoning models. https://developers.openai.com/api/docs/guides/reasoning?api-mode=responses, 2026. OpenAI API Documentation. Accessed: 2026-09-11.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Viacheslav Sinii, Alexey Gorbatovski, Artem Cherepanov, Boris Shaposhnikov, Nikita Balagansky, and Daniil Gavrilov. Steering LLM reasoning through bias-only adaptation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9202–9211, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.467. URL https://aclanthology.org/2025.emnlp-main.467/.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=HvoG8SxggZ.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage rl. 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, Main Conference, pp. 15181–15207. Curran Associates, Inc., 2025. doi: 10.52202/085713-0510. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/16371a9d5fed65d6d78ca3a7fa6e598c-Paper-Conference.pdf.

Tyler J VanderWeele. A three-way decomposition of a total effect into direct, indirect, and interactive effects. *Epidemiology*, 24(2):224–232, 2013.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors, 2025. URL https://arxiv.org/abs/2506.18167.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

Hanyu Wang, Bochuan Cao, Yuanpu Cao, and Jinghui Chen. TruthFlow: Truthful LLM generation via representation flow correction. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 62423–62444. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/wang25i.html.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Haolei Xu, Xinyu Mei, Yuchen Yan, Rui Zhou, Wenqi Zhang, Weiming Lu, Yueting Zhuang, and Yongliang Shen. EasySteer: A unified framework for high-performance and extensible llm steering. *arXiv preprint arXiv:2509.25175*, 2025.

Yang Xu, Kun Yao, Yiming Deng, Zheng Fang, Kai Ming Ting, and Ming Pang. Agpo: Asymmetric group policy optimization for verifiable reasoning and search ads relevance at jd, 2026. URL https://arxiv.org/abs/2605.05826.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation, 2026. URL https://arxiv.org/abs/2602.12125.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? 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. 57654–57689. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/537d5aa768c2d534016a4d06f87bc8fb-Paper-Conference.pdf.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Hf17y6u9BC.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. AdaptThink: Reasoning models can learn when to think. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3716–3730, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.184. URL https://aclanthology.org/2025.emnlp-main.184/.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jpxfof0EaS.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. 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, Main Conference, pp. 126546–126573. Curran Associates, Inc., 2025a. doi: 10.52202/085713-4220. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b8026fe01f5f8f7a80a1a9bf01d08cad-Paper-Conference.pdf.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=ftVlLG9cks.

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.

APPENDIX CONTENTS

A Extended Related Work **18**

B Extended Discussion **18**

B.1 Why does thinking leakage deserve attention? 18

B.2 Beyond linear response estimation. 18

C Experiment Setup **18**

D Supplementary Analysis for section 3 **20**

D.1 Control experiment: a reference axis 20

D.2 The leakage and reference axes across depth 22

D.3 Mode-gap geometry: quantities, construction, and fidelity 23

D.4 Where methods and models diverge 24

E Supplementary Analysis for section 4 **25**

E.1 Control experiment using v_{ref} 25

E.2 Steering at other depths 25

E.3 Why MiniCPM4.1-8B loses accuracy at the top dose. 27

E.4 Steering a post-trained checkpoint 27

E.5 The injected shift versus the shift the text carries 28

F Supplementary Analysis for section 5 **28**

F.1 Which checkpoints the analysis uses 28

F.2 Control experiment: reference and random axes 29

F.3 Doubling the dose 29

F.4 How much of the drift the leakage axis accounts for 31

F.5 Why counter-steering, and not a projection clamp 31

F.6 Counter-steering the base model 32

F.7 What counter-steering moves the model back to 32

F.8 The injected shift versus the shift the text carries 33

G Supplementary Analysis for section 6 **34**

G.1 Details of auditing on drift-induced gain 34

G.2 Question-level alignment 35

G.3 What the decomposition adds to the method comparison 35

G.4 An asymmetric response, and where post-training sits on it 36

G.5 Relation to three-way mediation decomposition 36

A Extended Related Work

Prior observations of thinking in NoThink. Work on post-training the NoThink mode has noted that the model often continues to think. Gan et al. (2026) call this reward hacking: a response is labeled NoThink because it begins with `</think>`, but continues with exploratory reasoning and collects the higher NoThink reward. They detect it from the text, by token usage and by the density of reasoning markers, the same behavioral signal we use in section 3. Zhang et al. (2025) cap NoThink responses at a fixed token budget to suppress it, and Gan et al. (2026) make the cap per-query; Tu et al. (2025), which imposes no such constraint, produces NoThink responses nearly as long as its Think ones. Zhu et al. (2025a) observe a related effect on Qwen3-4B in NoThink mode and read it the other way, as post-training unlocking latent reasoning ability. In a hybrid model that ability is already accessible through Think, so unlocking it in NoThink is not by itself an improvement in NoThink capability.

In these works leakage is incidental—an obstacle to remove, or a side benefit. We study it as the object of interest, and measure it in the model's internal representations as well as in its text.

B Extended Discussion

B.1 Why does thinking leakage deserve attention?

Post-training can elicit latent abilities from the base model (Yue et al., 2025; Karan & Du, 2026; Zhu et al., 2025a). Making previously inaccessible reasoning available can itself constitute a meaningful improvement, even without creating new abilities. Hybrid reasoning models present a different setting: the base model's reasoning behavior is already directly accessible through Think, where thinking leakage can inflate the apparent contribution of post-training by recruiting already accessible Think behavior into NoThink. A method's performance advantage may therefore reflect its greater effectiveness at re-invoking existing reasoning rather than the intended improvement within NoThink. The accuracy gain is real, but attributing it entirely to the latter overstates what the method achieved.

Thinking leakage does not preclude useful gains in reasoning efficiency. For example, our GRPO experiments use a generation budget of 8192 tokens, forcing the model to reach a correct final answer within a constrained reasoning budget. Post-training may thus make existing reasoning more concise even when its accuracy gains depend on leakage. For research aimed at raising the model's performance ceiling, however, such gains alone do not establish an advance beyond the reasoning already accessible through the base model's Think mode. This requires comparison with base Think across inference budgets to distinguish more efficient use of existing reasoning from a higher attainable level of performance.

B.2 Beyond linear response estimation.

Our causal decomposition does not require $Y(\theta, \pi)$ to be linear in either model weights or thinking engagement. The four outcomes in eq. (1) evaluate this function for the base model and a checkpoint, with and without steering or counter-steering. More densely sampled checkpoints would trace how this response changes along the training trajectory, while finer steering sweeps over a wider range would reveal its dependence on engagement. Together, these measurements could support a nonlinear response surface $Y(\theta, \pi)$, capturing finer-grained information. The same four-outcome accounting would apply to comparisons between selected points on this surface.

C Experiment Setup

Post-training methods. To test whether thinking leakage is specific to a particular training algorithm, we audit three post-training methods applied in NoThink:

- **GRPO** (Guo et al., 2025; Shao et al., 2024): Group Relative Policy Optimization method, for which we adopt the modified variant and configuration of Hübotter et al. (2026), trained on DAPO-Math (Yu et al., 2026);

- **SFT**: supervised fine-tuning on the reasoning traces of OpenThoughts-Math (Guha et al., 2025), following the configuration of Zhao et al. (2026);
- **OPSD**: on-policy self-distillation with or without LoRA (Zhao et al., 2026), using its original NoThink training recipe on OpenThoughts-Math (Guha et al., 2025). The LoRA configuration follows the original paper's setup while full parameter fine-tuning is also evaluated.

The detailed hyperparameter setting is given in table 1.

Table 1: Training and evaluation hyperparameters for the three fine-tuning recipes. Dashes denote parameters that do not apply to a given method.

Parameter	**GRPO**	**SFT**	**OPSD**
General			
Backbones	Qwen3-8B, Qwen3-4B, MiniCPM4.1-8B		
Training data	DAPO-Math	OpenThoughts-Math	OpenThoughts-Math
Data			
Max. prompt length	2048	2048	2048
Max. response length	8192	16000	1024
Optimization			
Optimizer	AdamW	AdamW	AdamW
Learning rate	1×10^{-6}	1×10^{-5}	5×10^{-6}
Warmup steps	10	0	0
Weight decay	0.01	0	0
Grad. clip norm	1.0	1.0	0.1
Effective batch size	32	32	32
Total steps	500	100	100
Evaluation frequency (/step)	100	25	25
Method-specific			
Rollout number	8	–	1
Rollout temperature	1.0	–	1.1
Normalize advantages	True	–	–
ϵ-high	0.28	–	–
KL coefficient	0	–	–
Per-token KL clip (threshold)	–	–	1×10^{-7}
Teacher	–	–	frozen θ_0
Distillation divergence	–	–	forward KL
LoRA rank / α	–	–	64 / 128 (if applied)
Evaluation			
Benchmarks	AIME24, AIME25, HMMT-Feb-25, HMMT-Nov-25		
Inference engine	vLLM		
Number of rollouts	16		
Temperature	0.6		
Top-p	0.95		
Max. generation length	32768 (63488 for MiniCPM4.1-8B)		

Evaluation data and protocol. We evaluate on 120 competition-math problems drawn equally from four benchmarks (30 each): **AIME24** (Zhang & Math-AI, 2024), **AIME25** (Zhang & Math-AI, 2025), **HMMT-Feb-2025**, and **HMMT-Nov-2025** (Dekoninck et al., 2026). For each problem we draw 16 samples (temperature 0.6, top-$p = 0.95$) and report the mean per-sample solve rate (*avg@16*). The per-sample binary score is the outcome $Y \in \{0, 1\}$ of section 2.1, and avg@16 is its empirical expectation $Y_{a,a'} = \mathbb{E}[Y]$. Full training and evaluation hyperparameters are listed in table 1.

Chunked generation and the loop filter. Rollouts vary widely in length, and in batched generation a batch is only as fast as its longest sequence, so generating everything one-shot at the maximum budget spends most of its time on a handful of stragglers. We therefore generate in chunks of 8192 tokens: after each round, only rollouts that reached that round's cap are continued, by re-feeding the

prompt and partial response, until they terminate or the budget is reached. Before each extension we re-test the accumulated text for degenerate repetition (word-level 12-gram repetition ratio above 0.3) and drop any rollout that fails from subsequent rounds.

Generation budgets. Both Qwen models use 32768 tokens in every setting. MiniCPM4.1-8B's THINK responses are far longer, so we raise both its base THINK run and its steering sweep—which induces long rollouts—to its context limit 63488 tokens, and keep 32768 elsewhere.

Correctness criterion. Each sample is graded binary and averaged over its 16 samples (avg@16). We extract the last `\boxed{}` expression as the final answer and compare it to the gold answer by exact string match, falling back to symbolic/ numeric equivalence via `math_verify` (Kydlíček, 2025). Responses with no parseable answer, including truncated or repetition-looped generations, are counted as incorrect.

Direction estimation. All directions $\hat{v}_{\text{leak}}$ and $\hat{v}_{\text{ref}}$ (section 2.2 and section D.1) are estimated on the *base* model θ_0. The estimation set comprises 120 problems $\times$ 16 rollouts.

Style tokens. The marker set is the union of two per-model lists, each collected from the vocabulary that model actually uses inside `<think>`. We refer to the two lists as M_{Qwen} and M_{MiniCPM} below. Matches are counted case-insensitively on word boundaries.

- M_{Qwen} (13): `wait`, `hmm`, `perhaps`, `maybe`, `actually`, `alternatively`, `seems`, `might`, `likely`, `check`, `reconsider`, `backtrack`, `instead`.
- M_{MiniCPM} (11): `since`, `similarly`, `moreover`, `notice`, `note`, `another`, `suppose`, `hence`, `recall`, `earlier`, `no`.

Taking the union matters because the two families are close to disjoint in practice — in THINK mode Qwen3 emits M_{Qwen} at 19 per 1000 words against 5 for MiniCPM4.1-8B, while MiniCPM4.1-8B emits M_{MiniCPM} at 19 against 12 — so scoring either model with the other's list alone would miss most of its markers.

Robustness of $\hat{v}_{\text{leak}}$ To assess the robustness of $\hat{v}_{\text{leak}}$, we re-estimate it on a *disjoint* set of 120 problems from the GRPO training data ($\times$1 rollout, 120 samples); the resulting direction remains closely aligned with the evaluation-set estimate at layer 20 (cosine 0.945), confirming that it is not an artifact of the evaluation distribution. It is equally insensitive to the sampling budget: subsampling a single rollout per evaluation problem recovers the $\times$16 direction with cosine 0.998 across all 16 such draws. Together, these show that $\hat{v}_{\text{leak}}$ is stable both across data distributions and across sampling budgets.

Interventions. Our causal experiments are performed by manipulating $\hat{v}_{\text{leak}}$ during NOTHINK generation—adding it in the base model (section 4) and subtract it in the post-trained model (section 5); implementation details are deferred to those sections.

Implementation. GRPO and SFT training were run with `verl` (Sheng et al., 2024),OPSD with the authors' public repository (Zhao et al., 2026), and the steering and counter-steering interventions with `EasySteer` (Xu et al., 2025). All experiments were run on two NVIDIA H200 GPUs.

D SUPPLEMENTARY ANALYSIS FOR SECTION 3

This appendix supports section 3 in four parts: a reference axis that tests whether the drift is specific to $\hat{v}_{\text{leak}}$ (section D.1), the same measurements repeated at three probe depths (section D.2), the quantities and construction behind the mode-gap figure (section D.3), and a breakdown of how the three post-training methods differ across the two model families section D.4).

D.1 CONTROL EXPERIMENT: A REFERENCE AXIS

To test whether the drift is specific to $\hat{v}^{(\ell)}_{\text{leak}}$, we repeat the measurement on a second axis built from the same rollouts.

A THINK rollout has two parts: the exploratory reasoning inside <think>...</think>, and the clean solution written after it. The leakage axis $\hat{v}^{(\ell)}_{\text{leak}}$ of eq. (3) contrasts NOTHINK against the rollout as a whole; contrasting it against the THINK solution region of $\mathcal{S}(x)$ alone gives

$$v^{(\ell)}_{\text{ref}} = \bar{h}^{(\ell)}_{\mathcal{S}}(\theta_0) - \bar{h}^{(\ell)}_{\mathcal{N}}(\theta_0), \qquad \hat{v}^{(\ell)}_{\text{ref}} = v^{(\ell)}_{\text{ref}} / \|v^{(\ell)}_{\text{ref}}\|. \tag{17}$$

The two axes separate what a checkpoint may be moving toward: drift along $\hat{v}^{(\ell)}_{\text{leak}}$ indicates exploratory reasoning leaking into NOTHINK, while drift along $\hat{v}^{(\ell)}_{\text{ref}}$ would indicate a shift toward the clean solution. A method that genuinely improved direct answering, without re-invoking deliberation, should only move along the latter.

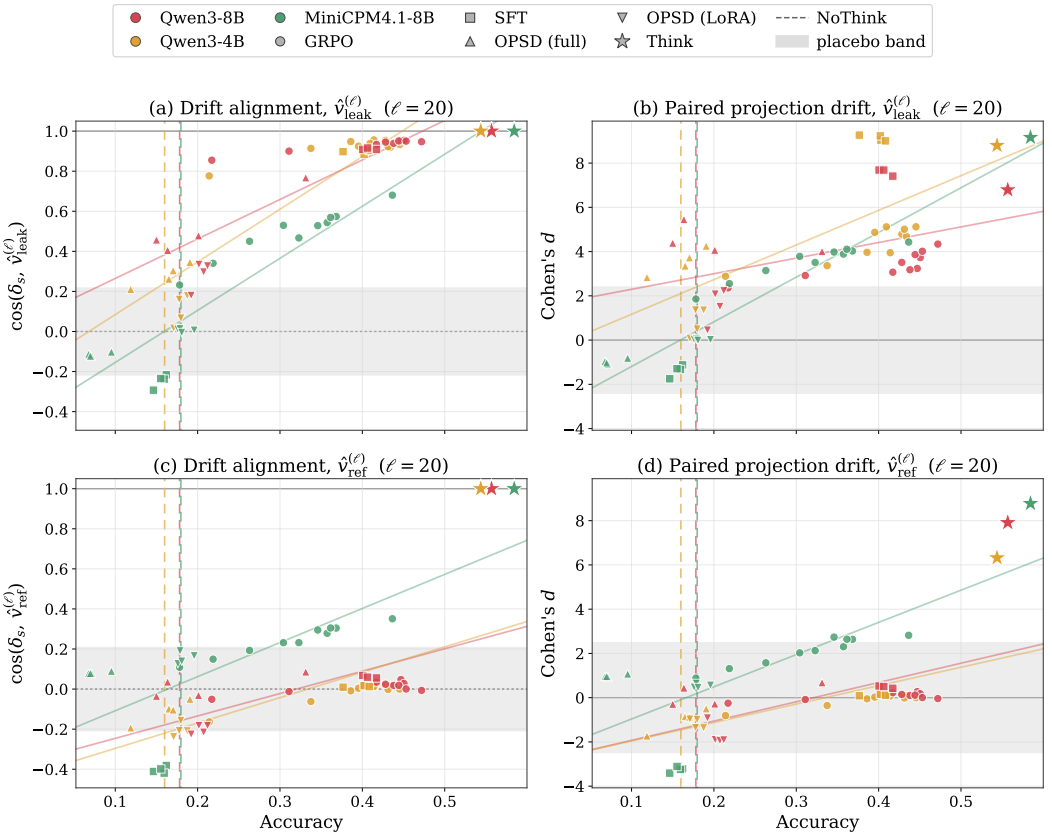


Figure 8: **The drift is specific to $\hat{v}^{(\ell)}_{\text{leak}}$ (layer $\ell = 20$).** Each point is one checkpoint; marker shape denotes the training method, colour the base model. $\star$ is the base model in THINK. **(a), (c)** Drift alignment on $\hat{v}_{\text{leak}}$ and $\hat{v}^{(\ell)}_{\text{ref}}$ respectively. **(b), (d)** Cohen's d of the same drift along each axis. The gray **placebo band** marks the null in each panel: in (a) and (c), the alignment obtained by splitting the base model's own NOTHINK rollouts in half, so that the drift carries sampling noise only; in (b) and (d), Cohen's d recomputed for the same checkpoint pair along a random unit direction. Both are drawn at the 90th percentile of the null's absolute value.

Result. The two axes behave differently in every respect (fig. 8). The drift alignment points squarely at $\hat{v}_{\text{leak}}$ (mean $\cos = 0.51$, 77% of checkpoints outside the null) and is unrelated to $\hat{v}_{\text{ref}}$ (0.00, 23%). The paired drift splits the same way: along $\hat{v}_{\text{leak}}$ it averages $d = 3.15 \pm 2.71$ with 66% of checkpoints clearing the placebo band, while along $\hat{v}_{\text{ref}}$ it is centred at zero (0.02 ± 1.37) with only 12% clearing it. Accuracy separates them as well ($\rho = +0.92$ against $+0.25$ for drift alignment, and $\rho = +0.68$ against $+0.23$ for paired drift).

Some drift along $\hat{v}_{\text{ref}}$ is expected—its anchor, the solution region, is part of the very rollout that anchors $\hat{v}_{\text{leak}}$, so the two cannot be fully independent—which makes the order-of-magnitude gap

the conservative reading. Post-training does not simply sharpen the model's ability to produce a clean solution; it re-invokes the THINK mode itself.

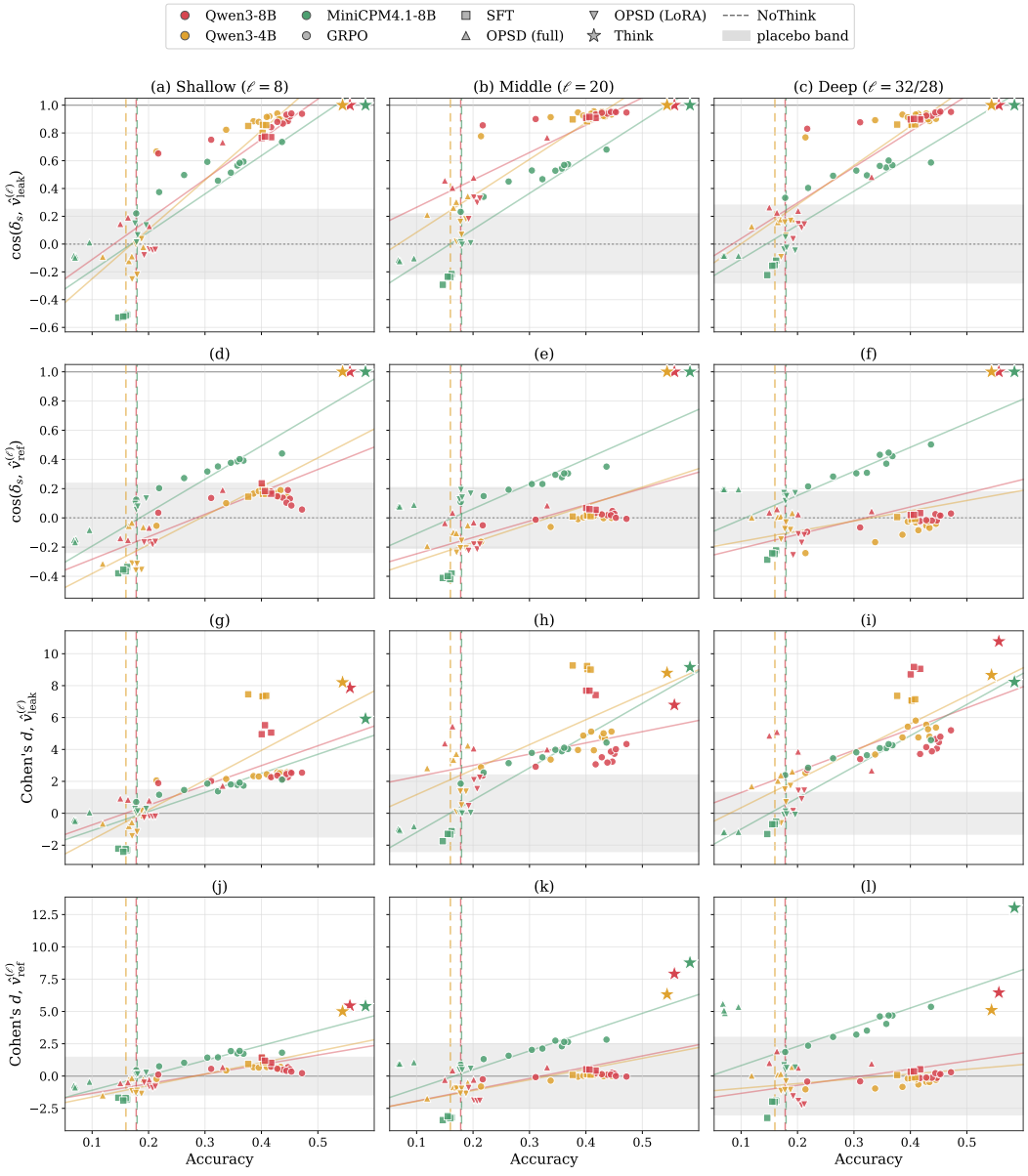


Figure 9: **The leakage happens across depth.** Rows are (measure, axis), columns are depth. Each point is one checkpoint; marker shape denotes the training method, colour the base model. The legend is the same as in fig. 8.

D.2 THE LEAKAGE AND REFERENCE AXES ACROSS DEPTH

To test whether the drift is universal to different layers, we repeat both measures on both axes at three probe depths: shallow $\ell = 8$ and middle $\ell = 20$ for all models, and a deep layer set to $\ell = 32$ for the 36-layer Qwen3 models and $\ell = 28$ for the 32-layer MiniCPM4.1-8B.

Result. Figure 9 reproduces the separation of section D.1 at every probe depth. Along $\hat{v}^{(\ell)}_{\text{leak}}$, 58–77% of checkpoints clear the placebo band on either measure, and drift alignment tracks NoThink accuracy for every model at every layer ($\rho = 0.83$–0.93); along $\hat{v}^{(\ell)}_{\text{ref}}$ only 12–29% clear it.

Depth scales the leakage effect without changing it: the paired drift roughly doubles from the shallow to the middle probe ($d = 1.49$ to 3.15) and is then unchanged at the deep layer (3.17). On the $\hat{v}^{(\ell)}_{\text{ref}}$, it stays near zero for the Qwen models at every depth ($|d| \leq 0.41$) and it grows with depth for the MiniCPM4.1-8B (0.23, 0.64, 2.35), where the two regions are least separable.

Table 2: Fidelity of fig. 3. *Drawn* is measured from the final two-dimensional coordinates; *true* is recomputed in $\mathbb{R}^{4096}$ from the raw activations. *Abs. err.* means absolute error and *Rel. err.* is the corresponding relative error.

Quantity	Checkpoint	True	Drawn	Abs. err.	Rel. err.
$\|v_{\text{leak}}\|$	base	15.426	15.426	-0.000	-0.00%
	GRPO@200	8.810	8.777	-0.034	-0.38%
	GRPO@500	7.178	7.142	-0.037	-0.51%
rotation vs. base	GRPO@200	$20.270°$	$19.555°$	$-0.715°$	-3.53%
	GRPO@500	$26.354°$	$25.844°$	$-0.510°$	-1.94%
$\|\bar{h}_{\mathcal{T}}\|$	base	72.318	73.525	$+1.207$	$+1.67\%$
	GRPO@200	72.149	73.525	$+1.376$	$+1.91\%$
	GRPO@500	71.927	73.525	$+1.598$	$+2.22\%$
$\|\bar{h}_{\mathcal{N}}\|$	base	76.975	78.515	$+1.540$	$+2.00\%$
	GRPO@200	74.429	73.087	-1.342	-1.80%
	GRPO@500	73.700	72.307	-1.393	-1.89%
ϕ	base	$11.312°$	$11.024°$	$-0.288°$	-2.54%
	GRPO@200	$6.658°$	$6.855°$	$+0.198°$	$+2.97\%$
	GRPO@500	$5.476°$	$5.532°$	$+0.056°$	$+1.02\%$

D.3 Mode-gap geometry: quantities, construction, and fidelity

D.3.1 Measured quantities

Following eq. (2), we compute the two mode centroids of each checkpoint at $\ell = 20$ in the model's native $d = 4096$ dimensions. We drop the layer and checkpoint indices below, writing $\bar{h}_{\mathcal{T}}, \bar{h}_{\mathcal{N}}$ for the centroids, $v_{\text{leak}} = \bar{h}_{\mathcal{T}} - \bar{h}_{\mathcal{N}}$ for their difference, and ϕ for the angle between them. Their values are listed in the *true* column of table 2.

Decomposing the change in $\|v_{\text{leak}}\|$. Writing $\|v_{\text{leak}}\|^2 = \|\bar{h}_{\mathcal{T}}\|^2 + \|\bar{h}_{\mathcal{N}}\|^2 - 2\|\bar{h}_{\mathcal{T}}\|\|\bar{h}_{\mathcal{N}}\| \cos\phi$, we recompute the base-to-GRPO@500 change holding one factor fixed. Closing ϕ at base norms gives $15.43 \to 8.54$, i.e. 84% of the observed drop; moving the norms at base ϕ gives $15.43 \to 14.46$, i.e. 12%. The remaining 4% is their interaction.

Which mode moves. Displacements from the base centroid, in units of the base gap $\|v_{\text{leak}}\| = 15.426$: $\bar{h}_{\mathcal{T}}$ moves 5.8% and 10.2% at steps 200 and 500, rotating $0.700°$ and $1.205°$; $\bar{h}_{\mathcal{N}}$ moves 53.2% and 66.2%, rotating $5.914°$ and $7.361°$. Think is effectively stationary under training while NoThink travels two thirds of the way to it, which is what gives the closing gap its direction.

D.3.2 Construction and fidelity of fig. 3

Figure 3 shows the two centroids, $\bar{h}_{\mathcal{T}}$ and $\bar{h}_{\mathcal{N}}$, and their difference in the same panel. These vectors live at very different scales: $\bar{h}_{\mathcal{T}}$ and $\bar{h}_{\mathcal{N}}$ have norm 75–77 while v_{leak} has norm 7–15, since $\bar{h}_{\mathcal{T}}$ and $\bar{h}_{\mathcal{N}}$ share a large common mode. No orthogonal projection can render both—the plane that preserves v_{leak} leaves the origin $78.0°$ outside it, drawing the centroid norms at 10%–20% of their

true length. We therefore use a approximate two-dimensional embedding rather than a projection. The three v_{leak} are placed exactly; the three $\bar{h}_{\mathcal{T}}$ are drawn as a single point; and the origin is then fitted by least squares to the four centroid norms. Table 2 compares the each drawn vector against its true value in $\mathbb{R}^{4096}$, showing that the plotting errors were kept within a narrow range.

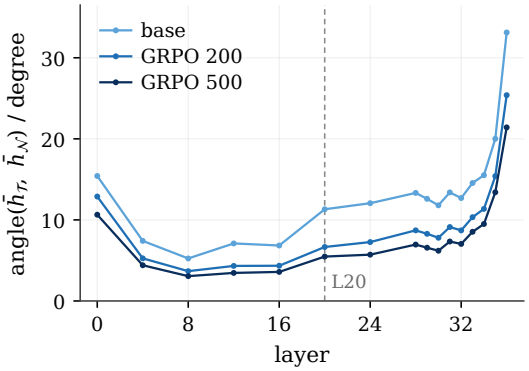


Figure 10: **The closing gap is a property of training, not of the probe layer.** Mode angle $\phi^{(\ell)}$ between $\bar{h}_{\mathcal{T}}^{(\ell)}$ and $\bar{h}_{\mathcal{N}}^{(\ell)}$ across all 16 probed layers of Qwen3-8B. Dashed line marks layer 20, used throughout the paper.

D.3.3 Depth robustness

Figure 10 repeats the mode angle $\phi^{(\ell)}$ at 16 probed layers. The ordering base $>$ GRPO@200 $>$ GRPO@500 holds at every depth, so the closing gap is a property of training and not of the layer at which we probe. The rise beyond layer 32 is the usual final-layer divergence as the residual stream is prepared for readout.

D.4 Where methods and models diverge

The three post-training methods differ in how far their training targets sit from the model's own distribution: GRPO trains on the model's own samples (on-policy), OPSD mixes self-generated and external traces (semi-on-policy), and SFT fits external traces outright (off-policy).

Off-policy training leaks more, and leaks differently. Comparing GRPO and SFT checkpoints at matched NoThink accuracy, the two methods separate on both Qwen models: SFT drifts roughly twice as far in paired effect size and carries about $1.5\times$ the marker density, and on 8B GRPO reaches slightly higher accuracy (0.423 against 0.408) while drifting less than half as far ($d = 3.29$ against 7.59). Alignment is high for both ($c \approx 0.90$–0.93), so the methods move in the same direction and differ only in how far.

Their trajectories differ as much as their endpoints. SFT has already reached its final drift at the earliest checkpoint we sample and stays there, whereas GRPO climbs monotonically across training. The two are consistent with different origins for the leakage: fitting external traces transfers the Think register wholesale within the first few steps, while on-policy RL accumulates it gradually, as the samples carrying Think-style computation are the ones that get rewarded. The semi-on-policy middle of the ordering cannot be read—OPSD and its LoRA variant barely lift accuracy above base, and their small drift is as consistent with a weak update as with a well-matched one.

Style mismatch drives both training failure and reversed drift. The corpus that SFT and OPSD train on is written in a measurable style, and that style matches one of the two model families but not the other (table 3): the traces favour M_{Qwen} markers over M_{MiniCPM} ones at a ratio of 1.72, close to what the two Qwen models produce in their own Think rollouts, while MiniCPM4.1-8B produces the opposite balance at 0.27.

On the Qwen3 models both methods train in the intended direction and both drift toward the $\hat{v}_{\text{leak}}$, though OPSD does so far less reliably than SFT, which reaches 0.417 on Qwen3-8B. At matched

Table 3: Style-token density per 1000 words, counted per word so that the three tokenizers are comparable.

Corpus	M_{Qwen}	M_{MiniCPM}	Ratio
OpenThoughts-Math(SFT, OPSD)	17.93	10.43	1.72
Qwen3-8B THINK	18.81	12.58	1.49
Qwen3-4B THINK	19.83	11.35	1.75
MiniCPM4.1-8B THINK	5.13	18.73	0.27

accuracy SFT drifts about twice as far as GRPO: fitting traces written in these models' own style is an efficient way to re-invoke THINK.

On MiniCPM4.1-8B neither method works well. Most SFT and OPSD checkpoints degrade in accuracy and drift away from the $\hat{v}_{\text{leak}}$. Both effects follow from the mismatch — the corpus asks the model to reason in a style it does not use. GRPO, trained on the model's own samples, is the only method that succeeds on all three models.

MiniCPM4.1-8B behaves differently from Qwen3 in the first place. The style-token balance is one of several ways the two families differ. Response length is another: MiniCPM4.1-8B's NOTHINK answers are $2.5\times$ longer than Qwen3's on the same problems, and its THINK rollouts $1.8\times$ longer. The architectures differ as well—32 layers against Qwen3's 36, so the middle layer we probe sits at a different relative depth. These base-model differences set the scale of what we measure—how many markers a rollout carries, how far the two mode centroids sit apart, how much a projection can move—so the magnitudes differ across families while the trends do not.

E SUPPLEMENTARY ANALYSIS FOR SECTION 4

This appendix supports section 4 in three parts: a control that repeats the intervention along the reference axis $\hat{v}_{\text{ref}}^{(\ell)}$ instead of v_{leak} (section E.1), the same sweep applied at six injection depths (section E.2), and an account of the accuracy drop MiniCPM4.1-8B shows at the largest dose (section E.3). Table 4 gives the full sweep from which the percentages quoted in the main text are computed. All tables in this subsection use the column definitions in table 4, unless otherwise noted.

E.1 CONTROL EXPERIMENT USING v_{ref}

To check that the steering result is specific to v_{leak}, we repeat it on the control axis v_{ref} (defined in eq. (17)), at ℓ=20 with α=1, on all three models (table 5). Note that, on all three models v_{ref} is the longer vector, so at a common α the control displaces the hidden state further than v_{leak} does.

On all the three models, neither accuracy nor reasoning-marker density moves off its base value, while the same dose along v_{leak} lifts both by a wide margin. This confirms leakage runs along v_{leak}. Response length does move, and it moves down. This says v_{ref} is not an empty direction but a meaningful one: it carries the concise clean-solution style of THINK mode after the exploratory reasoning inside `<think>...</think>`. Steering along it makes the model write a clean solution without doing the exploration that would normally produce one, so the imitation stays on the surface and accuracy does not rise.

E.2 STEERING AT OTHER DEPTHS

The steering results in the main text all inject $v_{\text{leak}}^{(\ell)}$ at ℓ=20 ($v_{\text{leak}}^{(20)}$). As a supplementary check we ask whether the effect is tied to that depth, or whether the $v_{\text{leak}}^{(\ell)}$ would works at other layers. We repeat the intervention on Qwen3-8B at six depths, $\ell \in \{12, 16, 20, 24, 28, 32\}$, estimating $v_{\text{leak}}^{(\ell)}$ separately at each layer following eq. (3) and holding everything else fixed.

The effect is present at every depth. Injecting at any of the six layers raises both NOTHINK accuracy and marker density above base (table 6). The leakage direction is therefore not an artifact

Table 4: **Full steering sweep results.** *Density* is the reasoning-marker density of section 3. *Trunc.* is the share of rollouts that reach the generation cap and *Loop* the share whose word-level 12-gram repetition ratio exceeds 0.3; the caps and the repetition threshold are those of section C.

Model	Configuration	Acc.	Density ($\times 10^{-3}$)	Trunc. (%)	Loop (%)	Len. (tok)
Qwen3-8B	base, NoThink	0.178	3.08	3.1	7.9	4,268
	base, Think	0.557	18.30	12.4	1.1	17,782
	GRPO@500 (best ckpt)	0.472	11.85	1.6	2.2	9,293
	steering, v_{leak} *at* ℓ=20					
	$\alpha = 0.5$	0.231	5.57	1.3	6.0	4,940
	$\alpha = 1$	0.335	10.26	0.8	2.9	6,850
	$\alpha = 1.25$	0.374	13.07	0.9	2.7	8,603
	$\alpha = 1.5$	0.435	15.32	1.2	2.6	10,773
	$\alpha = 2$	0.451	17.75	6.0	2.7	14,587
Qwen3-4B	base, NoThink	0.160	3.24	2.3	7.8	4,069
	base, Think	0.544	18.37	10.0	0.7	17,552
	GRPO@450 (best ckpt)	0.445	10.66	0.3	1.6	8,912
	steering, v_{leak} *at* ℓ=20					
	$\alpha = 0.5$	0.215	5.89	1.1	7.1	4,828
	$\alpha = 1$	0.299	11.62	0.3	2.9	6,570
	$\alpha = 1.25$	0.352	13.96	2.0	2.1	8,647
	$\alpha = 1.5$	0.396	15.60	1.0	2.1	10,234
	$\alpha = 2$	0.422	16.28	4.4	0.9	14,352
MiniCPM4.1-8B	base, NoThink	0.180	3.76	5.1	31.7	10,268
	base, Think	0.584	10.55	5.8	0.4	31,430
	GRPO@500 (best ckpt)	0.436	7.30	0.4	1.6	8,812
	steering, v_{leak} *at* ℓ=20					
	$\alpha = 0.5$	0.280	5.02	2.0	9.7	14,291
	$\alpha = 1$	0.398	7.21	2.8	1.2	18,315
	$\alpha = 1.25$	0.440	8.23	3.7	0.6	22,100
	$\alpha = 1.5$	0.461	9.49	4.9	0.4	25,995
	$\alpha = 2$	0.421	11.02	14.0	2.6	32,669

Table 5: **Steering along the reference axis instead of the leakage axis** (α=1, ℓ=20).

Model	Configuration	Acc.	Density ($\times 10^{-3}$)	Trunc. (%)	Loop (%)	Len. (tok)
Qwen3-8B	base, NoThink	0.178	3.08	3.1	7.9	4,268
	steer along v_{leak}	0.335	10.26	0.8	2.9	6,850
	steer along v_{ref}	0.159	3.10	0.1	1.9	2,057
Qwen3-4B	base, NoThink	0.160	3.23	2.3	7.8	4,069
	steer along v_{leak}	0.299	11.62	0.3	2.9	6,570
	steer along v_{ref}	0.150	3.00	0.0	2.1	1,950
MiniCPM4.1-8B	base, NoThink	0.180	3.76	5.1	31.7	10,268
	steer along v_{leak}	0.398	7.21	2.8	1.2	18,315
	steer along v_{ref}	0.148	3.57	1.4	4.0	5,463

of one probe layer: the same contrast, estimated independently at each depth, moves the model toward Think wherever it is applied.

It is strongest at ℓ=20, on every measure. ℓ=20 gains twice as much accuracy as the next best depth, and is the only one to push marker density past three times the base rate. ℓ=20 is also the most stable, with the lest degenerate repetitions compared with other layers. Injecting deeper still produces markers but buys almost no accuracy—the same split between surface signature and

capability that the dose sweep shows, now along depth. A coarse scan over the other two models points the same way, with ℓ=20 again the best depth on both.

Table 6: **Steering the same direction at different depths (Qwen3-8B, α=1).** $v^{(\ell)}_{\text{leak}}$ is injected at one layer at a time; †marks ℓ=20, the layer used everywhere else in the paper.

Injection layer	Acc.	Density ($\times 10^{-3}$)	Trunc. (%)	Loop (%)	Len. (tok)
base, NoThink	0.178	3.08	3.1	7.9	4,268
$\ell = 12$	0.254	4.25	2.1	5.6	6,527
$\ell = 16$	0.233	4.27	1.3	7.3	5,358
$\ell = 20^{\dagger}$	0.335	10.26	0.8	2.9	6,850
$\ell = 24$	0.224	6.68	0.5	4.2	4,260
$\ell = 28$	0.208	6.33	1.0	8.9	4,473
$\ell = 32$	0.201	5.52	1.5	10.9	4,467

E.3 Why MiniCPM4.1-8B loses accuracy at the top dose.

MiniCPM4.1-8B is the only model whose accuracy drops at the largest dose, from 0.461 at α=1.5 to 0.421 at α=2, while the share of rollouts that hit the 63488-token cap rises from 4.9% to 14.0% (table 4). A truncated rollout produces no `\boxed` answer and is scored wrong, so we look at where both fall across questions.

Truncation is spread out rather than stuck on a few bad items: at α=2 it reaches just over half the questions, and on most of those only a small share of the 16 rollouts is affected. It does fall mainly on the hard questions, which average 0.29 accuracy at α=1.5 against 0.69 for the questions with no truncation. The accuracy drop sits in the same place: on the questions where nothing is truncated, the two doses score the same, so the whole drop comes from questions where some rollout ran out of tokens.

Table 7: **Comparison of steered checkpoint with the base model.** The steered row adds α=0.5, 1, 1.5 of v_{leak} to the best GRPO checkpoint of Qwen3-8B.

Configuration	Acc.	Density ($\times 10^{-3}$)	Trunc. (%)	Loop (%)	Len. (tok)
base, NoThink	0.178	3.08	3.1	7.9	4,268
base, Think	0.557	18.30	12.4	1.1	17,782
GRPO@500	0.472	11.85	1.6	2.2	9,293
+ steering, α=0.5	0.545	16.42	1.3	1.8	12,179
+ steering, α=1	0.553	18.31	3.8	1.0	15,033
+ steering, α=1.5	0.537	18.40	7.7	0.9	17,102

E.4 Steering a post-trained checkpoint

The steering experiments in the main text start from the base model. Here we apply the same operator, at the same layer, to a model that has already been post-trained: Qwen3-8B's best GRPO checkpoint (table 7).

Post-training does not use up the axis, and the ceiling is Think itself. Adding v_{leak} to the checkpoint raises accuracy from 0.472 to 0.545 at α=0.5 and to 0.553 at α=1, where it comes within 0.4 percents of what the base model reaches in Think mode under the same generation budget, with the marker density arriving at the same place (18.31 against Think's 18.30). Pushing further does not help: at α=1.5 density no longer moves and accuracy falls back. Training and steering therefore compose rather than substitute, and the base model's Think behavior acts as the ceiling—the direction still has room in it after training, but NoThink behavior converges on Think rather than passing it.

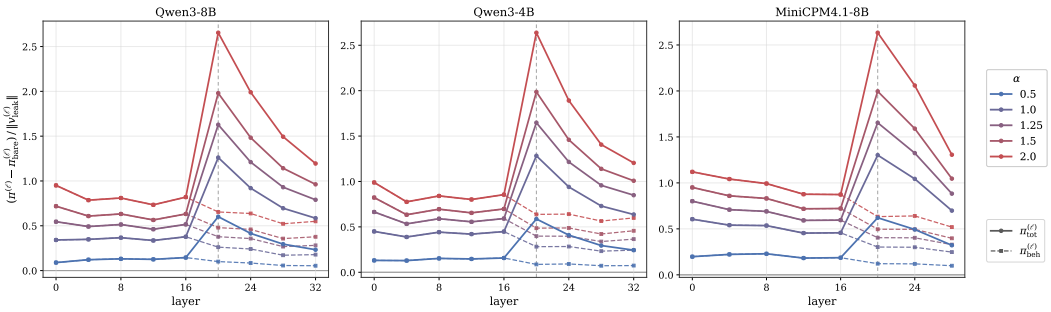


Figure 11: **Depth profile of steering** ($\alpha = 0.5, 1, 1.25, 1.5, 2$). $\pi^{(\ell)}$ across probe depths, relative to each model's bare NoThink run and divided by $\|v^{(\ell)}_{\text{leak}}\|$. Colour is the steering dose. *Total drift* ($\pi^{(\ell)}_{\text{tot}}$, solid) is measured during steered generation; *behavioral drift* ($\pi^{(\ell)}_{\text{beh}}$, dashed) re-forwards the same text without the intervention. The dashed vertical line marks the injection layer ℓ=20.

E.5 The injected shift versus the shift the text carries

After the injection, the total drift $\pi^{(\ell)}_{\text{tot}}$ mixes two things: the steering vector we add, $\alpha\|v^{(20)}_{\text{leak}}\|$, and the influence of the steered tokens. To isolate the latter, we re-forward each steered generation without the intervention, so the resulting profile, $\pi^{(\ell)}_{\text{beh}}$, reflects only the text.

Both $\pi^{(\ell)}_{\text{tot}}$ and $\pi^{(\ell)}_{\text{beh}}$ grow with dose, but behave differently with depth (fig. 11). The total shift peaks at the injection layer and decays over the following layers, as later computation absorbs part of the added vector. The behavioral drift $\pi^{(\ell)}_{\text{beh}}$ is smaller and present at every depth, including layers below the injection point: re-forwarding adds nothing anywhere, yet passing the steered text through the model displaces it along v_{leak} from the first layers on.

F Supplementary Analysis for Section 5

This appendix supports section 5 in four parts. Section F.1 sets out the grouping of the fifteen checkpoints on which the analysis rests, and section F.2 repeats the intervention along a reference and a random axis. Four variants of the operator follow: twice the dose, the whole drift rather than its leakage component, a projection clamp, and the base model in place of a checkpoint (sections F.3 to F.6). The last two read what the intervention leaves behind—comparing the counter-steered checkpoint to the base model question by question (section F.7), and separating the injected shift from the shift the generated text carries (section F.8). Table 8 gives the per-checkpoint results from which the numbers quoted in the main text are computed. Throughout this section, all tables use the column definitions in table 4 unless otherwise noted. Shading follows table 8 and indicates the checkpoint groups defined in section F.1.

F.1 Which checkpoints the analysis uses

Counter-steering removes a checkpoint's drift along v_{leak}, so what it can tell us depends on what training put there. The fifteen checkpoints fall into three groups on two measurements that are fixed before any intervention is run: how far the checkpoint's accuracy rose above its NoThink base, and how the training drift is oriented with respect to v_{leak} (table 8).

Aligned (nine checkpoints). Training moved these models a long way onto the leakage axis and made them better NoThink solvers ($\cos \geq 0.53$, gain $\geq +0.12$). Both a gain and a mechanism that could carry it are present, so these are the checkpoints the analysis in section 5 mainly uses.

No usable gain (four checkpoints). Training left these models no better than their base, whether or not it moved them along the axis ($|\text{gain}| \leq 0.04$, $0.01 \leq \cos \leq 0.46$). With no gain there is

Table 8: **Full counter-steering result** (ℓ=20, γ=1). *bare* is the original checkpoint, *c.-s.* the same checkpoint under counter-steering. $c = \cos(\delta_s, \hat{v}_{\text{leak}})$ as defined in eq. (6). Rows are grouped as described in section F.1: unshaded, the nine *aligned* checkpoints; grey, the four with *no usable gain*; blue, the two *reverse-drift* checkpoints.

			Accuracy		Density		Loop (%)		Len. (tok)	
Model	Checkpoint	c	bare	c.-s.	bare	c.-s.	bare	c.-s.	bare	c.-s.
Qwen3-8B	GRPO@500	+0.95	0.472	0.349	11.85	6.51	2.2	4.5	9,293	6,991
	GRPO@200	+0.95	0.447	0.315	9.47	4.55	4.3	13.9	12,583	9,936
	SFT@50	+0.91	0.417	0.267	13.91	9.72	2.4	14.2	13,745	12,337
	OPSD@25	+0.77	0.331	0.210	6.59	2.70	4.8	28.2	13,847	9,244
	OPSD@100	+0.46	0.150	0.131	2.33	1.90	23.3	28.0	8,613	6,088
Qwen3-4B	GRPO@450	+0.93	0.445	0.312	10.66	5.45	1.6	6.8	8,912	7,187
	GRPO@200	+0.96	0.414	0.284	9.94	5.01	0.6	12.1	10,667	8,021
	SFT@75	+0.90	0.408	0.261	14.08	9.83	1.3	19.3	14,035	12,840
	OPSD@50	+0.35	0.191	0.152	2.26	1.83	29.7	34.3	7,299	6,409
	OPSD@25	+0.21	0.119	0.102	1.38	1.14	50.4	51.9	6,068	5,638
MiniCPM4.1-8B	GRPO@500	+0.68	0.436	0.304	7.30	5.78	1.6	6.6	8,812	6,575
	GRPO@200	+0.53	0.304	0.243	6.83	4.97	9.8	25.2	8,257	7,299
	SFT@100	-0.22	0.162	0.179	5.21	5.50	46.0	38.5	11,091	11,850
	OPSD-LoRA@75	+0.01	0.196	0.189	3.92	4.01	26.8	28.6	10,540	10,662
	OPSD@50	-0.11	0.068	0.085	2.60	2.61	15.1	13.0	12,108	13,360

nothing whose drift-induced gain can be tested, and no denominator to normalize a loss against, so they are excluded.

Reverse drift (two checkpoints). Training moved these models *reversely* on the leakage axis, and they did not improve ($\cos \leq -0.11$, gain ≤ -0.02). These are used as a complementary group of the aligned group.

F.2 Control experiment: reference and random axes

To confirm that the counter-steering result of section 5 is specific to v_{leak}, we repeat it on each model's best checkpoint along two control axes: the reference axis $\hat{v}_{\text{ref}}$ of section D.1, and a direction drawn at random per question. The dosing rule is unchanged—γ=1 subtracts the checkpoint's drift component along whichever axis is used—so the control dose is not chosen by us but read off the drift itself, and a null result can arise two ways: because training put nothing on that axis, or because what it put there does not matter.

On the Qwen models, training put nothing there. Along $\hat{v}_{\text{ref}}$ the drift components are 0.07 and 0.001, against drift lengths of 10.52 and 5.57 and below the 0.16 and 0.11 a random direction picks up by chance (table 9). The prescribed dose is therefore negligible, and both controls leave accuracy and marker density where they were.

On MiniCPM4.1-8B the dose is real, and still costs nothing. Here the $\hat{v}_{\text{ref}}$ component is about half the v_{leak} one, so the control is a substantive intervention rather than a no-op. It leaves accuracy and marker density unchanged, while removing the v_{leak} component of the same drift costs roughly a third of the accuracy. This is the stronger form of the control: what a checkpoint gains in training sits on v_{leak}, and displacing it by a comparable amount along another meaningful axis does not touch the gain.

F.3 Doubling the dose

Counter-steering at γ=1 nominally returns a checkpoint to the base engagement level. At γ=2 it travels the same distance again, landing as far below π_0 as the checkpoint sat above it.

Table 9: **Counter-steering along the leakage axis and two control axes** (ℓ=20, γ=1, best checkpoint per model).

Model	Configuration	Acc.	Density ($\times 10^{-3}$)	Trunc. (%)	Loop (%)	Len. (tok)
Qwen3-8B	GRPO@500	0.472	11.85	1.6	2.2	9,293
	counter-steering					
	v_{leak}	0.349	6.51	1.2	4.5	6,991
	v_{ref}	0.459	12.06	0.5	2.1	9,090
	random	0.455	12.05	0.9	1.9	9,325
Qwen3-4B	GRPO@450	0.445	10.66	0.3	1.6	8,912
	counter-steering					
	v_{leak}	0.312	5.45	1.0	6.8	7,187
	v_{ref}	0.435	10.61	0.4	1.4	9,097
	random	0.430	10.64	0.5	1.9	9,145
MiniCPM4.1-8B	GRPO@500	0.436	7.30	0.4	1.6	8,812
	counter-steering					
	v_{leak}	0.304	5.78	0.4	6.6	6,575
	v_{ref}	0.451	7.36	0.8	1.5	9,119
	random	0.449	7.24	0.3	1.4	8,807

Table 10: **Comparison of counter-steering at $\times 1$ and $\times 2$ dose** (γ=1, 2).

		Accuracy		Density		Loop (%)		Len. (tok)	
Model	Checkpoint	$\times 1$	$\times 2$	$\times 1$	$\times 2$	$\times 1$	$\times 2$	$\times 1$	$\times 2$
Qwen3-8B	GRPO@500	0.349	0.212	6.51	3.43	4.5	13.3	6,991	6,078
	GRPO@200	0.315	0.191	4.55	2.40	13.9	22.6	9,936	7,696
	SFT@50	0.267	0.131	9.72	4.79	14.2	30.1	12,337	9,587
	OPSD@25	0.210	0.087	2.70	1.21	28.2	59.6	9,244	6,154
	OPSD@100	0.131	0.076	1.90	1.53	28.0	41.4	6,088	5,353
Qwen3-4B	GRPO@450	0.312	0.212	5.45	2.82	6.8	12.0	7,187	5,799
	GRPO@200	0.284	0.191	5.01	2.74	12.1	15.9	8,021	6,387
	SFT@75	0.261	0.118	9.83	4.66	19.3	36.5	12,840	9,776
	OPSD@50	0.152	0.119	1.83	1.31	34.3	39.4	6,409	5,525
	OPSD@25	0.102	0.086	1.14	1.14	51.9	56.0	5,638	5,174
MiniCPM4.1-8B	GRPO@500	0.304	0.228	5.78	4.69	6.6	29.3	6,575	6,349
	GRPO@200	0.243	0.187	4.97	4.27	25.2	44.5	7,299	7,248
	SFT@100	0.179	0.201	5.50	6.34	38.5	33.4	11,850	12,788
	OPSD-LoRA@75	0.189	0.191	4.01	4.04	28.6	27.5	10,662	10,451
	OPSD@50	0.085	0.098	2.61	2.64	13.0	8.9	13,360	13,355

Each group extends its own trend. Doubling the dose continues what γ=1 started (table 10). In the aligned group accuracy and marker density fall again, ending close to the checkpoints' own NoThink base on both measures. The reverse-drift group moves the other way at both doses: these checkpoints drifted backward along v_{leak}, so subtracting their drift adds the direction rather than removing it, and accuracy rises with dose.

This gives a two-sided control on the intervention. Most checkpoints drifted toward Think along v_{leak}, and counter-steering subtracts that drift, so it removes the direction; a few drifted backward, and subtracting their negative drift adds the direction instead. The operator is the same in both cases. If the accuracy change came from disrupting the model, both groups would get worse. Instead the aligned group loses accuracy and the reverse-drift group gains it, each following the sign of its own drift.

Why we do not use γ=2 quantitatively. Repetition climbs steeply in the aligned group and ends above the base model the checkpoints were pushed back to, so part of the accuracy drop here is

degenerate generation rather than removed leakage, and the two cannot be separated. (In the reverse-drift group repetition falls with dose, so the rise is not a mechanical consequence of adding more dose.) Second, γ=2 is try to reduce π below base NoThink level, where the base model no longer serves as a reference point: at γ=1 we can ask whether accuracy returns to base, but past it the state is one no version of the model ever generated from. We therefore read this dose as evidence that the effect is graded and direction-dependent, and base no quantitative claim on it.

Table 11: **Removing the $\hat{v}_{\text{leak}}$ component of the drift against removing the whole drift** (Qwen3-8B, ℓ=20). The $\hat{v}_{\text{leak}}$ arm is the counter-steering used throughout the paper, which removes $\Delta\pi\, v_{\text{leak}}$ with $\gamma = 1$; the δ_s arm removes the drift in full.

Configuration	Acc.	Density ($\times 10^{-3}$)	Trunc. (%)	Loop (%)	Len. (tok)
GRPO@500	0.472	11.85	1.6	2.2	9,293
$-\ \hat{v}_{\text{leak}}$	0.349	6.51	1.2	4.5	6,991
$-\ \delta_s$	0.325	6.18	1.6	5.6	6,571
GRPO@200	0.447	9.47	6.2	4.3	12,583
$-\ \hat{v}_{\text{leak}}$	0.315	4.55	5.2	13.9	9,936
$-\ \delta_s$	0.308	4.75	3.9	9.4	8,945

F.4 How much of the drift the leakage axis accounts for

Counter-steering using $\Delta\pi\, v_{\text{leak}}$ removes only the component of the drift that lies along $\hat{v}_{\text{leak}}$, and fig. 3 shows there is another component orthogonal to it. The drift $\delta_s = \bar{h}_{\mathcal{N}}(\theta_s) - \bar{h}_{\mathcal{N}}(\theta_0)$ decomposes as $\delta_s = \Delta\pi\, v_{\text{leak}} + v_{\perp}$. To check whether $v_{\perp}$ matters, we remove δ_s in full instead.

The axis accounts for almost all of it. As shown in table 11, removing δ_s in full reduces only slightly more accuracy than removing its leakage-axis component alone, so the axis captures 84% and 95% of what the full intervention achieves. The axis estimated on the base model therefore remains the right one after post-training.

Table 12: **Counter-steering versus the projection clamp**, averaged over the 15 checkpoints of section 5. Both set γ=1.

Configuration	Acc.	Density ($\times 10^{-3}$)	Trunc. (%)	Loop (%)	Len. (tok)
base, NoThink	0.173	3.36	3.5	15.8	6,202
post-trained checkpoint	0.304	7.22	4.9	14.7	10,391
+ counter-steering	0.226	4.77	3.7	21.7	8,963
+ projection clamp	0.240	5.84	3.6	24.5	9,220

F.5 Why counter-steering, and not a projection clamp

Counter-steering shifts the projection by a constant, which sets the mean engagement but leaves each position free to depart from it. The operation that realises $\mathrm{do}(\pi{=}\pi_0)$ exactly is to pin the coordinate at every position instead,

$$h_t^{(\ell)} \ \leftarrow\ h_t^{(\ell)} - \Big(\big\langle h_t^{(\ell)}, \hat{v}_{\text{leak}}^{(\ell)} \big\rangle - \pi_0^{(\ell)}\Big)\hat{v}_{\text{leak}}^{(\ell)}, \tag{18}$$

which forces the projection to the base level exactly, and removes the token-to-token variation in engagement along with it.

Counter-steering removes more and breaks less. At the same dose, counter-steering gives both lower marker density and lower accuracy than the clamp, and produces fewer degenerate repetitions (table 12). We attribute the clamp's shortfall to the variance it flattens: holding every position at

one projection costs coherence rather than leaked computation. Neither operator is clean—counter-steering also raises the repetition rate above the original checkpoint, so part of the accuracy it removes is disruption rather than leakage. The difference suggests the measurement is still limited by the operator: a finer intervention, one that reaches the same engagement level while staying within states the model could have produced on its own, should remove more of the leaked behavior at less collateral cost, and so read the leakage level more accurately.

F.6 COUNTER-STEERING THE BASE MODEL

Applying counter-steering to the base models reverses the pattern seen under base steering: accuracy, marker density and response length all fall, and the loop rate rises (table 13).

The response below base is much weaker than above it. Per unit of imposed $\Delta\pi\, v_{\text{leak}}$, base counter-steering loses 0.064, 0.060 and 0.109 on Qwen3-8B, Qwen3-4B and MiniCPM4.1-8B—roughly half the 0.147, 0.133 and 0.215 that base steering gains per unit in the opposite direction, and half the 0.166, 0.178 and 0.212 that counter-steering costs a checkpoint. Moving the base model below its own π_0 buys far less change than moving it above.

Table 13: **Counter-steering the base model** (ℓ=20). The magnitude is set as $\gamma\|v_{\text{leak}}\|$ since there is no drift for a base model.

Model	Configuration	Acc.	Density ($\times 10^{-3}$)	Trunc. (%)	Loop (%)	Len. (tok)
Qwen3-8B	base, NoThink	0.178	3.08	3.1	7.9	4,268
	$\gamma = 0.5$	0.151	1.99	1.3	8.6	3,322
	$\gamma = 1$	0.111	1.33	1.9	9.5	3,233
Qwen3-4B	base, NoThink	0.160	3.24	2.3	7.8	4,069
	$\gamma = 0.5$	0.129	2.02	1.0	10.4	3,270
	$\gamma = 1$	0.100	1.48	0.8	10.1	2,877
MiniCPM4.1-8B	base, NoThink	0.180	3.76	5.1	31.7	10,268
	$\gamma = 0.5$	0.119	3.14	1.6	54.9	8,209
	$\gamma = 1$	0.074	2.71	1.7	70.9	7,620

F.7 WHAT COUNTER-STEERING MOVES THE MODEL BACK TO

Counter-steering costs accuracy and adds repetition, and a larger dose does more of both. Two readings fit that: the intervention is damaging the model in a new way, or it is undoing what training did and returning it toward the base model. Since every question is answered 16 times, we can tell these apart question by question: if the intervention is damaging, its per-question pattern should follow neither the checkpoint nor the base model; if it is undoing training, the pattern should shift from the checkpoint toward the base model as the dose grows.

The repetition pattern resembles the base model's. Which questions repeat under counter-steering is predicted by which questions repeat in the base model ($\rho = +0.52$ at γ=1, $+0.64$ at γ=2), and the match is closer at the larger dose. The two sets also overlap: 84% of the questions that repeat under counter-steering also repeat in the base model.

Accuracy moves the same way, more slowly. Base and checkpoint question-level accuracy are themselves correlated across questions ($\rho = +0.76$), since both follow how hard each question is, so we compare using partial correlations that hold one of them fixed. At γ=1 the counter-steered accuracy stays closer the checkpoint more than the base model ($+0.78$ against $+0.43$); at γ=2 the two are level ($+0.50$ against $+0.58$). Accuracy moves toward the base model as the dose grows, but not as far as repetition does.

The repetition does not drive the accuracy loss. Keeping only rollouts that terminate normally, accuracy still falls from 0.459 to 0.342 at γ=1 and to 0.269 at γ=2, which is 91% and 82% of the full loss. What counter-steering removes is therefore not just the ability to finish a response. The

exception is MiniCPM4.1-8B at γ=2, where repetition reaches 29% of rollouts and accounts for two fifths of the loss; there the accuracy result cannot be read on its own.

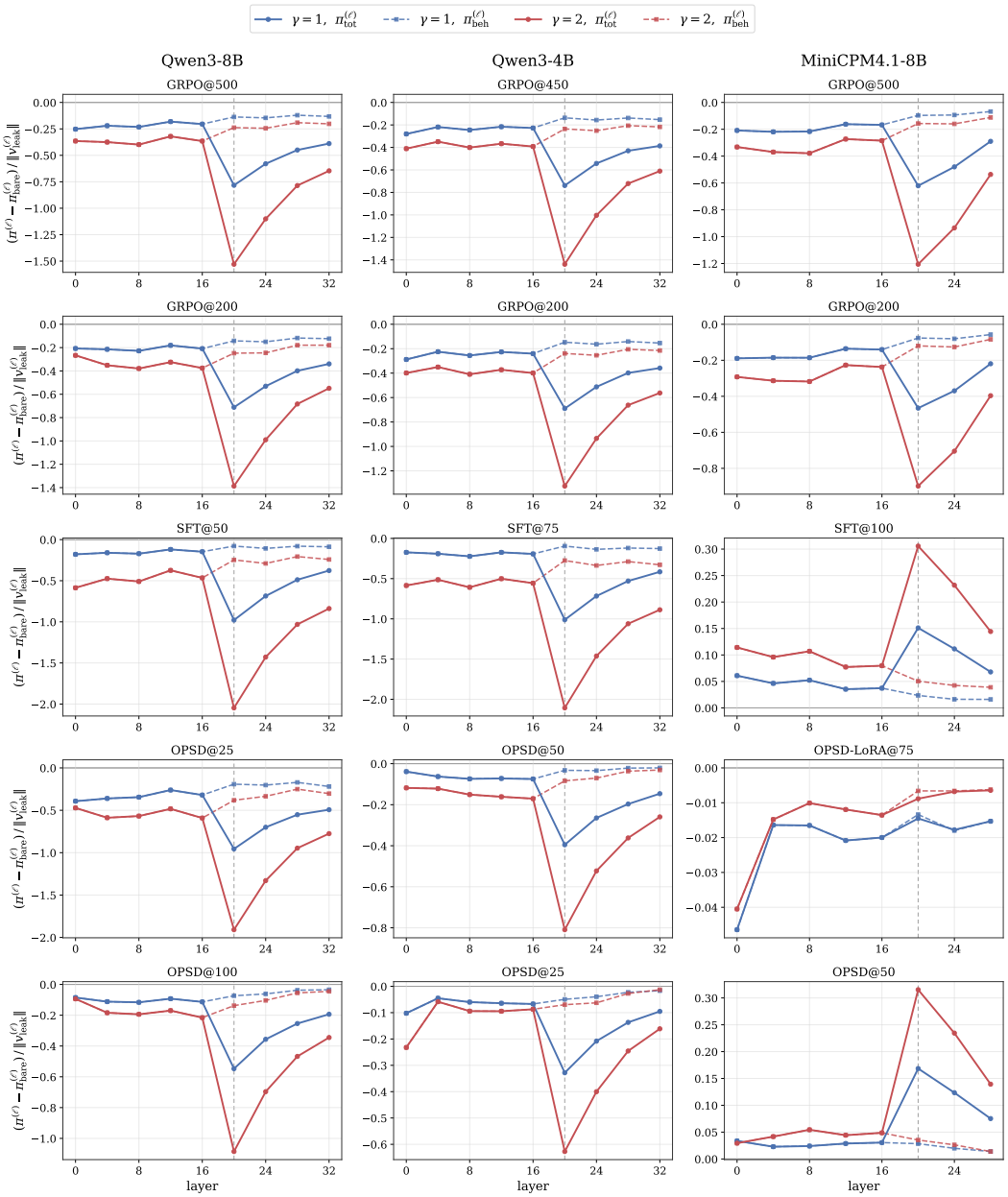


Figure 12: **Depth profile of counter-steering, checkpoint by checkpoint** ($\gamma = 1, 2$). Columns are models, rows their five checkpoints. In each panel, π^{ℓ} across probe depths, relative to that checkpoint's original run and divided by $\|v^{(\ell)}_{\text{leak}}\|$. Blue is γ=1, red γ=2; *Total drift* ($\pi^{(\ell)}_{\text{tot}}$, solid) is measured during counter-steered generation; *behavioral drift* ($\pi^{(\ell)}_{\text{beh}}$, dashed) re-forwards the same text without the intervention. The dashed vertical line marks the injection layer ℓ=20.

F.8 The injected shift versus the shift the text carries

Figure 12 repeats the decomposition of section E.5 on the counter-steering side, per checkpoint. As defined in section E.5, $\pi^{(\ell)}_{\text{tot}}$ denotes the total drift and $\pi^{(\ell)}_{\text{beh}}$ is the behavioral drift. The picture mirrors it: the two curves (solid and dashed) coincide below the injection layer, the gap that opens

there closes with depth, and $\pi^{(\ell)}_{\text{beh}}$ stays away from zero throughout, so part of the engagement change is again carried by the tokens.

The panels also split by the sign of $\Delta\pi$. Where a checkpoint drifted backward during training, the operator adds v_{leak} rather than subtracting it, and both curves move up instead of down. These are the same checkpoints whose accuracy rises rather than falls under the operator (section F.3).

G Supplementary Analysis for section 6

This appendix supplements section 6 in five parts. Section G.1 reports the per-checkpoint gain decompositions and interprets the leakage ratio across checkpoint groups. Section G.2 examines whether the aggregate effects align at the question level. Section G.3 extends the training method comparison using the decomposition. Section G.4 proposes an asymmetric response profile that would account for the steering and counter-steering slopes, and section G.5 relates the bidirectional audit to a three-way mediation decomposition as a theoretical extension.

Table 14: **Per-checkpoint gain decompositions** ($\ell = 20$). Notation follows section 6.

		via the base sweep		via counter-steering			
Model	Checkpoint	L_{base}	R_1	L_{trained}	R_0	$\mathcal{A}$	λ
Qwen3-8B	GRPO@500	+0.095	+0.199	+0.123	+0.171	+0.028	0.418
	GRPO@200	+0.084	+0.185	+0.132	+0.136	+0.049	0.492
	SFT@50	+0.132	+0.107	+0.150	+0.089	+0.018	0.627
	OPSD@25	+0.112	+0.041	+0.121	+0.032	+0.009	0.789
	OPSD@100	+0.070	-0.098	+0.019	-0.047	-0.051	-0.667
Qwen3-4B	GRPO@450	+0.080	+0.205	+0.133	+0.153	+0.053	0.465
	GRPO@200	+0.072	+0.182	+0.130	+0.124	+0.057	0.510
	SFT@75	+0.122	+0.126	+0.147	+0.102	+0.025	0.591
	OPSD@50	+0.048	-0.018	+0.039	-0.008	-0.009	1.271
	OPSD@25	+0.037	-0.078	+0.017	-0.058	-0.021	-0.405
MiniCPM4.1-8B	GRPO@500	+0.113	+0.144	+0.133	+0.124	+0.020	0.517
	GRPO@200	+0.084	+0.041	+0.061	+0.063	-0.022	0.494
	SFT@100	-0.027	+0.010	-0.017	-0.001	+0.011	0.941
	OPSD-LoRA@75	+0.000	+0.016	+0.007	+0.009	+0.007	0.419
	OPSD@50	-0.030	-0.082	-0.018	-0.094	+0.012	0.158

G.1 Details of auditing on drift-induced gain

Table 14 gives the per-checkpoint decomposition along both paths. We read it by the three groups of section 5, which are fixed before any intervention is run.

Aligned checkpoints. Both terms are positive throughout: the training improved NoThink accuracy ($\Delta > 0$), part of that improvement is induced by drift on $\hat{v}_{\text{leak}}$ ($L_{\text{trained}} > 0$), and part survives it ($R_0 > 0$). Thus λ here takes its plainest reading, the fraction of drift-induced gain, and runs from 0.42 to 0.79. The ordering follows the training method: the GRPO checkpoints sit near 0.5, SFT near 0.6, and OPSD@25 on Qwen3-8B highest at 0.79.

No usable gain. The three Qwen OPSD checkpoints in this group share a pattern: L_{trained} is positive while R_0 is negative. Counter-steering still removes accuracy, so the drift did carry a genuine effect, but the improvement outside the drift is negative. Since $\Delta = L_{\text{trained}} + R_0$ is then small or negative, λ falls outside $[0, 1]$ and is no longer meaningful. The remaining checkpoint, OPSD-LoRA@75 on MiniCPM4.1-8B, has both terms positive but both within noise ($L_{\text{trained}} = +0.007$, $R_0 = +0.009$): that run barely moved the model in any direction.

Reverse drift. Both terms are negative. The negative L_{trained} says the reverse drift decrease accuracy, and counter-steering recovers the decreasing. The negative R_0 says the rest of the training hurt as well. Because both change sign together with Δ, λ stays in $[0, 1]$ and keeps its meaning: on

MiniCPM4.1-8B SFT@100 it reads 0.94, so almost all of that checkpoint's degradation is explained by the reverse drift.

G.2 QUESTION-LEVEL ALIGNMENT

In this subsection, we examine whether THINK, post-training, and the two interventions affect similar questions, beyond their aggregate accuracy effects.

For each of the 120 questions, let $a_{w,c}(q)$ denote mean correctness over 16 rollouts, where $w = 0, 1$ denotes the base model or trained checkpoint and c specifies the mode or intervention. We define

$$\begin{aligned} g_{\mathrm{T}}(q) &= a_{0,\textsc{think}}(q) - a_{0,\textsc{nothink}}(q), \\ g_{\mathrm{S}}(q) &= a_{0,\mathrm{steer}}(q) - a_{0,\textsc{nothink}}(q), \\ \Delta(q) &= a_{1,\textsc{nothink}}(q) - a_{0,\textsc{nothink}}(q), \\ L_{\mathrm{trained}}(q) &= a_{1,\textsc{nothink}}(q) - a_{1,\mathrm{counter}}(q). \end{aligned} \tag{19}$$

Here, steering uses the fixed dose $\alpha = 1$, while counter-steering uses the checkpoint-specific full dose $\gamma = 1$. Table 15 reports Spearman correlations across questions.

Question-level agreement across interventions. Across the nine aligned checkpoints, post-training gains Δ correlate positively with thinking gains g_{T} ($\rho = 0.54$–0.89), steering gains g_{S} ($\rho = 0.57$–0.79), and counter-steering losses L_{trained} ($\rho = 0.59$–0.82). Questions that benefit more from post-training therefore tend to benefit more from THINK and steering, and to lose more under counter-steering. Non-aligned checkpoints show weaker correlations involving post-training gains, consistent with their limited training effectiveness. Together with the positive alignment between thinking and steering gains ($\rho(g_{\mathrm{T}}, g_{\mathrm{S}}) = 0.61$–$0.73$), these results extend the aggregate findings in sections 3 to 6 to individual questions.

Table 15: **Question-level alignment of gains and intervention effects.** Spearman correlations across 120 questions. The base-only correlation $\rho(g_{\mathrm{T}}, g_{\mathrm{S}})$ is identical within each model. Shading follows table 8.

Model	Checkpoint	$\rho(g_{\mathrm{S}}, \Delta)$	$\rho(\Delta, L_{\mathrm{trained}})$	$\rho(g_{\mathrm{T}}, \Delta)$	$\rho(g_{\mathrm{T}}, g_{\mathrm{S}})$
Qwen3-8B	GRPO@500	+0.772	+0.766	+0.836	+0.677
	GRPO@200	+0.792	+0.806	+0.852	
	SFT@50	+0.754	+0.752	+0.882	
	OPSD@25	+0.572	+0.643	+0.677	
	OPSD@100	+0.341	+0.510	+0.349	
Qwen3-4B	GRPO@450	+0.741	+0.820	+0.874	+0.613
	GRPO@200	+0.730	+0.798	+0.837	
	SFT@75	+0.677	+0.818	+0.885	
	OPSD@50	+0.289	+0.503	+0.274	
	OPSD@25	-0.253	+0.237	-0.085	
MiniCPM4.1-8B	GRPO@500	+0.793	+0.812	+0.737	+0.732
	GRPO@200	+0.601	+0.588	+0.540	
	SFT@100	+0.100	+0.426	+0.141	
	OPSD-LoRA@75	+0.336	+0.575	+0.294	
	OPSD@50	-0.269	+0.409	-0.018	

G.3 WHAT THE DECOMPOSITION ADDS TO THE METHOD COMPARISON

Section D.4 compares the three post-training methods by how far they drift along $\hat{v}_{\mathrm{leak}}$. The decomposition lets us ask the same question in units of accuracy.

On Qwen3-8B the checkpoints that gained and drifted onto the axis have L_{trained} of $+0.123$ (GRPO@500), $+0.150$ (SFT@50) and $+0.121$ (OPSD@25). What the three methods buy along the leakage direction is therefore comparable. Their R_0 is not: $+0.171$, $+0.089$ and $+0.032$ over the same three checkpoints. The methods differ mainly in what they achieved *outside* the drift, and on-policy RL is the one that achieved most of it.

This also explains why λ orders the methods the way it does. OPSD@25 has the largest λ (0.79) not because it leaked more—its L_{trained} is the smallest of the three—but because its total gain is roughly half, so almost nothing it achieved lies outside the drift. A large λ can come from a large numerator or a small denominator, and here it is the latter.

G.4 An asymmetric response, and where post-training sits on it

Section 6 finds β_{base} and β_{trained} close to each other. Base counter-steering (section F.6) is the one case that does not match: its slope is roughly half of either.

Figure 13 is the reading we favour. The accuracy response along $\hat{v}_{\text{leak}}$ is not a single slope but a curve, asymmetric about the base model's own position: displacements above π_0 are worth about twice as much as displacements below it, where the response compresses. Post-training does not rebuild this curve, it moves the model along it. A checkpoint therefore sits on the steep branch, which is why counter-steering a checkpoint and steering the base model—both operating above π_0—return matching slopes, while counter-steering the base does not.

Counter-steering at $\gamma = 2$ supports the same picture (section F.3): it moves a checkpoint from $+\Delta\pi$ through π_0 down to $-\Delta\pi$, so the second half of the trip lies in the compressed region ($\pi < \pi_0$) and costs less than the first half.

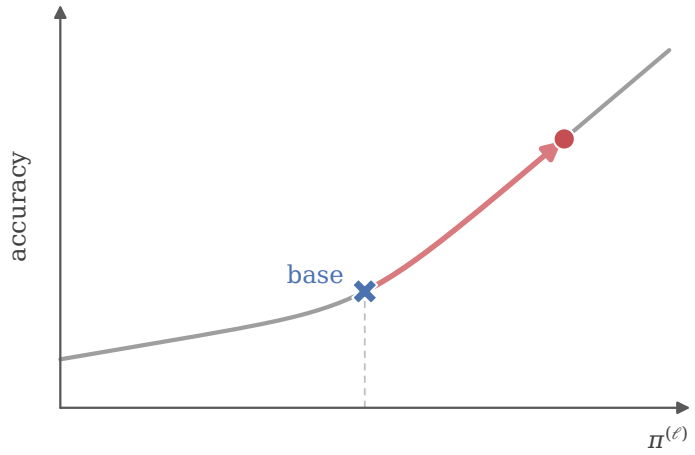


Figure 13: **A schematic showing that the response along $\hat{v}_{\text{leak}}$ is asymmetric about the base model's position, and post-training moves the model along it.**

G.5 Relation to three-way mediation decomposition

The two paths in eq. (12) can be combined into a three-way accounting of the total gain:

$$\Delta = \underbrace{R_0}_{\text{remaining gain}} + \underbrace{L_{\text{base}}}_{\text{base steering effect}} + \underbrace{\mathcal{A}}_{\text{interaction}} . \tag{20}$$

This parallels the algebraic structure of the three-way mediation decomposition (VanderWeele, 2013), with $\mathcal{A} = L_{\text{trained}} - L_{\text{base}}$ capturing the difference in intervention effects between the base and trained checkpoints.

When $\mathcal{A}$ is small, the two paths yield similar readings of the gain. Larger asymmetry would indicate that the effect of displacement along $\hat{v}_{\text{leak}}$ depends more strongly on the checkpoint. The three-way form makes this dependence explicit, providing a useful extension for comparing models or training methods with more divergent intervention responses. We present this formulation as a theoretical extension of the current decomposition eq. (12), offering a framework for future studies of how post-training changes the model's response to leakage.