BIDIRECTIONAL TRANSFER, ASYMMETRIC EXPRESSION: MATH FINE-TUNING CONCEALS THE CODE GAINS IT CARRIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Transfer between mathematics and code is usually scored as net change on a target benchmark against the base model. Reports disagree even on its sign, yet all read that change as what the update contains. We find instead that an update carries two things, the capability it has learned and the answering mode it adopts at its first token, and the latter can mask the former across domains. We call this masking *concealed transfer* and measure it against a format-matched control, the base plus the update rows that carry format. Against this control, transfer on Qwen3-4B runs both ways but asymmetrically: code fine-tuning raises mathematics directly, whereas mathematics fine-tuning shows no detectable difference on code. This apparent absence is not an absence of transfer. An edit fixed before evaluation and blind to target scores lifts the code score above both the fine-tuned model and the control. The same concealment recurs at three checkpoints across two architectures, each with its own releasing edit. On Qwen3-4B the concealment traces to one direction in weight space carrying a vanishing share of the update's energy. Removing it restores code openings and releases the gain with no measurable source-domain cost. Random directions of matched rank and energy do neither, and this direction alone does not shift the opening. These findings recast cross-domain transfer as separating what a model can solve from how it begins to answer, and an undetected net difference does not establish zero transfer.

1 INTRODUCTION

Large language models need to serve multiple task domains, so whether gains learned in one domain benefit another governs both the composition of training data and the value of merging specialist models. Studies of this question assess transfer through changes in a target domain benchmark score, measured against the base model (Huan et al., 2026; Krishna et al., 2025) or against a model trained without the source-domain data (Ma et al., 2024; Zhang et al., 2025). Reports on transfer between mathematics and code disagree even on its sign, with Dong et al. (2024) reporting performance conflicts when mixing mathematics, code, and general data under supervised fine-tuning in high-resource settings, Huan et al. (2026) reporting negative transfer under supervised fine-tuning, Krishna et al. (2025) reporting positive transfer from code to mathematics but negative transfer in the reverse direction, and Hu et al. (2026) reporting positive transfer in both directions. Despite these conflicting directions, all of these reports assume that a change in net score reveals what the update contains, and none uses a format-matched control as its reference. This paper provides that control.

That measurement fails as soon as an update changes how the model answers while it adds domain ability, because the added ability reaches the net score only through a compatible answering genre. Genre is the mode of answering that the model commits to from its opening token, whether it reasons in natural language first or writes the program directly when given a code problem (Figure 1). Format is the surface convention the answer obeys, such as code fences, language tags, and stopping. For the same problem, whether the model reasons first or writes the program first is a matter of genre, and whether it wraps the code in a fence is a matter of format. The control of Section 2 restores format

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

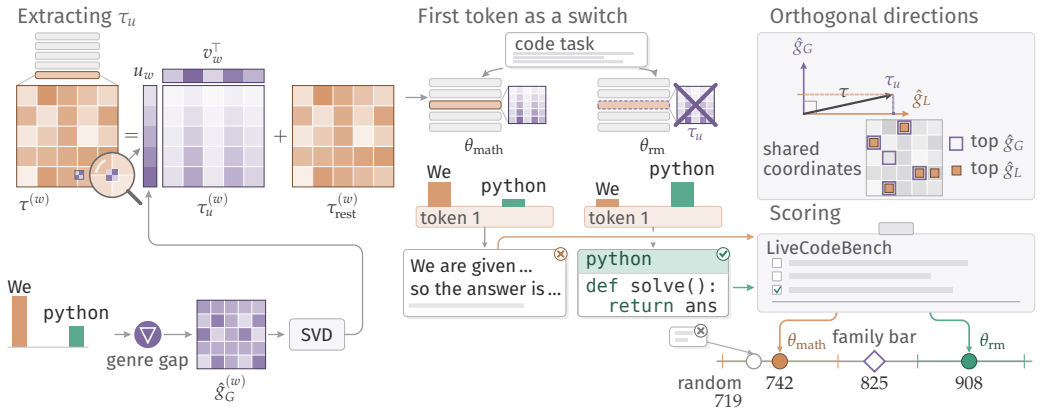

Figure 1: A rank-one direction in the update sets the first token. Each of 252 write matrices splits $\tau = \tau_u + \tau_{\text{rest}}$, τ_u (1.3×10^{-5} of $\|\tau\|^2$) from the first-token genre-gap gradient. Removing it turns 1,004/1,055 math openings into 0, lifting LiveCodeBench from 742 to 908 above the threshold 825. A random direction gives 719. $\hat{g}_G \perp \hat{g}_L$ (cosine -0.077), same coordinates (Spearman 0.78).

alone. Section 4 then isolates, from the update and before any target-domain score is consulted, a single direction that changes genre and nothing else, using only the genre tendency at the opening.

Relative to the control, the mathematics fine-tuned model scores 742 versus 766 on LiveCodeBench, a difference of -24 with a 95% interval $[-74, 27]$ that includes zero and indicates no detectable difference. Yet the update is not empty, as removing a direction that changes the opening genre raises LiveCodeBench to 908, 142 above the control, without reducing mathematics accuracy. The undetected difference in net score reflects how the model answers, while the gain carried by the update remains concealed. This paper calls a set of measurements that jointly satisfies three conditions *concealed transfer* (Section 2). The mathematics update moves the openings on the vast majority of code problems from the code genre to the mathematics genre, a change termed *genre shift*.

The contributions of this paper are as follows. With the control that restores the rows carrying format as the reference, transfer is bidirectional but its expression is asymmetric, and the eight base families resolve into concealment, direct expression, or a source domain that does not rise above the noise band (Section 3, Table 1). Because the net score measures the ability already expressed by a checkpoint, a format-matched control and an edit fixed before scoring, blind to target scores and requiring no training, distinguish unexpressed transfer, termed concealed transfer, from absent transfer (Sections 2 and 3). Concealment recurs at three checkpoints across two architectures, including a public Qwen3-14B checkpoint, and at each an edit fixed before scoring and blind to target-domain scores releases the gain. The direction along genre is decomposed only on Qwen3-4B. On rnj-1 it is located but not decomposed, and on Qwen3-14B the releasing edit leaves the opening unchanged (Section 5). In the Qwen3-4B LoRA update, genre shift is tied to one direction defined by the genre logit gradient at the first generated position and carrying roughly one part in 100,000 of the update energy, and removing this direction eliminates mathematics openings with no detectable source domain decline, whereas a random direction of the same rank and energy has neither effect (Section 4).

2 Preliminaries and Setup

Models and updates. Table 1 lists the eight base families (LLM-Core Xiaomi, 2025; Team GLM, 2024; Essential AI, 2025; Bakouch et al., 2025; Team Olmo, 2025; Tencent Hunyuan, 2025). The discovery family, Qwen3-4B-Base (Qwen Team, 2025), is fine-tuned on a NoThink mathematics set and, for the reverse direction, on a row-matched code set. Every family update is trained with LoRA on the same data and templates (Appendix A), while the full-parameter Qwen3-14B checkpoint was trained and released by Huan et al. (2026) (Section 5). Let θ_0 denote the public base, θ_s the model after supervised fine-tuning on source domain s, and $\tau = \theta_s - \theta_0$ the update, or task vector (Ilharco et al., 2023). The update is partitioned by the same rule in every family into a shared block a,

comprising the few embedding and output-projection rows that actually change (five for Qwen3-4B, with per-family counts in Appendix B), and a linear block τ_{lin}, comprising the entire update to the attention and MLP projections in every decoder layer (Appendix B).

Control. The control is $\theta_0 + a$, the base with only the shared block added back. Its mathematics avg@8 is 43 against 31 for the base, a difference of $+12$ with interval $[-2, 28]$, and on LiveCodeBench the base with a code-block prefix comes within 17 of it (749 against 766), so format alone accounts for the control's code score. It shares the fine-tuned model's surface conventions while its decoder layers are those of the base. Reporting format and ability separately, and fine-tuning every compared model on the same task data before evaluation, both have precedent (Zhou et al., 2023; Lin et al., 2023; Dominguez-Olmedo et al., 2025).

Evaluation. The target benchmark for mathematics to code is the 1,055-problem LiveCodeBench v6 (Jain et al., 2025) with an 8k token budget, while the target benchmark for code to mathematics is MATH500 (Hendrycks et al., 2021; Lightman et al., 2024). The source benchmarks are AIME 2025, AIME 2026, and the February 2026 HMMT, evaluated by pass@1 avg@8 with an 8k budget and 16k as a sensitivity check. The secondary code benchmarks are CRUXEval-O (Gu et al., 2024) and HumanEval+ (Liu et al., 2023; Chen et al., 2021), while MBPP+ (Liu et al., 2023; Austin et al., 2021) is used to estimate the direction in Section 4. Appendix A specifies the number of correct attempts k, sampling, decoding, sandbox execution, and intervals.

Criteria. The noise band is the 95% interval for the paired difference between arms with the problem as the cluster, following the clustering and pairing of Miller (2024) and computed by bootstrap. The source domain "rises above the noise band" when the interval for the fine-tuned model relative to the control is positive and excludes zero, while the target domain "falls within the noise band" when that interval contains zero. A set of measurements constitutes concealed transfer when the source domain rises above the noise band; the target domain falls within it; and an intervention fixed before scoring, constructed without target domain scores or training, raises the model above both the fine-tuned model and the control. The intervention must also clear the family threshold, computed against the control by the max-T procedure of Westfall & Young (1993) over the family's edit arms, and on Qwen3-4B the single edit fixed before scoring is held to the threshold computed over the 24 coarse-grained arms (Appendix A). At all three checkpoints the arm that clears this threshold also scores above the fine-tuned model.

3 Bidirectional transfer and asymmetric expression

This paper measures transfer as the difference between the fine-tuned model and the control. Qwen3-4B provides complete measurements in both directions, so the analysis begins with this family and then tests the remaining bases under the same conditions. Table 1 reports the scores and intervals, with each family's control construction and interval convention given in the table notes.

3.1 Code gain bypasses the genre

The gain from code to mathematics needs no carrier and lands directly on the net score. Code fine-tuning of Qwen3-4B has 141 more correct MATH500 attempts than the control, interval $[105, 179]$. None of the 1,500 answers opens in the code genre, and at position 1 the code update pushes genre with the opposite sign to the mathematics update. Removing one direction along the genre gradient from this code update, by the construction of Section 4, changes MATH500 by 9 relative to the fine-tuned model, an interval $[-11, 30]$ that includes zero. A random direction of the same dose also yields an interval that includes zero. The opening does not drift and the score does not move: on this side, the gain has nothing to do with genre.

Most of the decline on Hunyuan-7B occurs on problems answered with a code fence, with no detectable difference outside fenced answers (Appendix J). For Qwen3-4B, the code-to-mathematics gain does not pass through genre.

Table 1: Family differences from the shared-block control.

Family	Source Δ [CI]	Target Δ [CI] (math to code, LCB)	Score of the clearing edit	Reverse Δ [CI] (code to math, MATH500)	Outcome
Qwen3-4B	$+77$ $[46, 110]$	-24 $[-74, 27]$	908	$+141$ $[105, 179]$	concealment
Qwen3-8B	$+82$ $[49, 118]$	$+212$ $[161, 262]$	none	$+343$ $[304, 384]$	direct expression
rnj-1	$+39$ $[19, 63]$	$+31$ $[-24, 84]$	946	$+383$ $[335, 436]^*$	concealment
SmolLM3-3B	-38 $[-64, -17]$	$+83$ $[44, 122]$	none	-260 $[-303, -216]$	source not above noise band
Hunyuan-7B	$+17$ $[-11, 45]$	$+249$ $[196, 302]$	none	-64 $[-102, -27]$	source not above noise band
MiMo-7B	$+105$ $[69, 146]$	$+524$ $[462, 589]$	none	$+290$ $[250, 336]$	direct expression
OLMo-3-7B	$+20$ $[-6, 46]$	$+272$ $[227, 318]$	none	-113 $[-159, -69]$	source not above noise band
GLM-4-9B	$+12$ $[3, 22]$	$+79$ $[34, 125]$	none	$+227$ $[184, 271]$	direct expression
Qwen3-14B†	$+142$ $[98, 187]$	-25 $[-79, 30]$	1,134	—	concealment

LCB is LiveCodeBench. The control is $\theta_0 + a$, the base plus the shared block, with row counts by family in Appendix B. Intervals are paired cluster-bootstrap 95% intervals with the problem as the cluster and B=2000. Thresholds are 825 (Qwen3-4B), 734 (rnj-1), and 982 (Qwen3-14B). *rnj-1 reverse reference is θ_0, since $\theta_0 + a$ returns empty output on 477/477/470 of 500. †External replication, whose source cell pools MATH500 and avg@8 (Appendix L). Hunyuan-7B source cell at 16k (Appendix K).

3.2 Net score unchanged, update not empty

Since the code update's gain bypasses the opening, the question is whether the mathematics update does the same, with the source domain switched to mathematics and the target to code on the same base. Against the base, the mathematics fine-tuned model rises from 548 to 742 on LiveCodeBench, with 18.6% of the base's answers truncated against 3.5% for the control, an apparent success. Against the control, 742 stands against 766, a difference of -24 with interval $[-74, 27]$ that includes zero: there is no detectable difference. The two reference points yield two verdicts, and most of the 194-attempt gap is format, since prefixing the base with a single code block reaches 749, close to the control without any training. The reference point should therefore be the format-matched control, and against it mathematics fine-tuning shows no detectable difference in the code net score.

The update is nonetheless not empty. It raises CRUXEval-O from 27 for the control to 4,667 across 8,000 attempts, a rise that conflates format and ability but rules out zero gain on code. LiveCodeBench provides stronger evidence, as removing a direction that changes the opening genre raises the score to 908, 142 above the control, without reducing mathematics accuracy (Section 4), while adding a code-block prefix at the opening raises the fine-tuned model from 742 to 894 across 3,165 attempts but moves the control from 766 to 781. The prefix releases 15 from the control (766 to 781) and 152 from the fine-tuned model (742 to 894), so what the prefix unlocks is carried by the update and not by the prefix. The outcomes for the remaining bases determine whether this contrast between an unchanged net score and a substantive update is specific to Qwen3-4B.

3.3 Distribution of concealment across eight base families

Concealment is not universal. All eight base families are evaluated with the same data, templates, control construction, and threshold procedure, and Table 1 reports the resulting measurements and intervals. The source domain is evaluated with an 8k budget, and six of the seven families that produce overlong answers are reevaluated at 16k, which changes the outcome for Hunyuan-7B because its source domain difference contracts to 17 attempts, an interval $[-11, 45]$ that includes zero (Appendix K).

The eight families yield different outcomes. Qwen3-4B and rnj-1 exhibit concealment, with each edit clearing its family threshold at 908 versus 825 and 946 versus 734. Qwen3-8B, MiMo-7B, and GLM-4-9B express the gain directly, exceeding the control by 212, 524, and 79 with intervals $[161, 262]$, $[462, 589]$, and $[34, 125]$, while full-parameter fine-tuning of Qwen3-4B on the same data also expresses the gain directly at 198 above the control with interval $[147, 252]$. For SmolLM3-3B

and OLMo-3-7B, and for Hunyuan-7B under the longer budget, the source domain does not rise above the noise band.

3.4 Genre shift as the visible feature of concealment

Where there is concealment there is genre shift, but not the converse. The opening genre moves from code to mathematics, the genre shift of Section 1. Under the argmax reading on Qwen3-4B, the fine-tuned model opens 1,004 problems in the mathematics genre and the control opens none. Per-family counts for the other families are in Table J.2 in the appendix.

Every sampled MiMo-7B opening is in the mathematics genre even though its target domain rises above the noise band with its source domain, while Qwen3-14B releases the gain after removal of the MLP part despite retaining mathematics openings. Genre shift is therefore a visible feature of concealment that can identify families worth further testing, but the component that carries concealment must be located within the update (Section 4). Section 4 therefore examines the update for a component that changes answer genre without carrying the gain itself.

4 One direction in the update carries the genre shift

The most direct assumption is that training pushes the base too far and genre is a byproduct of that excess, in which case reducing the dose should restore genre while retaining capability. The code-block prefix is an inference-time intervention that presupposes the target genre. The direction below lies inside the update without that presupposition, and what the two share is the object this section looks for. Linear interpolation between the base and fine-tuned weights is a standard way to adjust the dose of an update (Wortsman et al., 2022), and proportional scaling performs exactly this interpolation. At $\lambda = 0.628$, scaling raises LiveCodeBench from 742 to 876 and clears the threshold of 825 while leaving more than a quarter of the mathematics openings, whereas at $\lambda = 0.12$, where those openings reach zero, the score falls to 798, showing that genre and capability do not vary in step with dose. Scaling an update linearly by layer depth and suppressing shallow layers can mitigate forgetting while preserving target-task performance (Wang et al., 2025a), but although this makes dose layer dependent, the rule remains blind to the function of each scaled component. If genre and capability occupied different modules, retaining one module's update should recover one without the other. Instead, retaining either the attention part or the MLP part clears the threshold on Qwen3-4B, at 856 and 870 despite their 23% and 77% shares of update energy, while on Qwen3-14B the arm that retains the attention part is the one that clears the threshold (Section 5.2). Module-level treatment is well motivated in the merging literature, where submodules are substantially more linear than the full model and fine-tuning the attention modules improves weight disentanglement in task arithmetic (Dai et al., 2025; Jin et al., 2025).

Modules provide the coarsest notion of location, while individual parameter coordinates provide the finest. Skills acquired through fine-tuning can be localized to as little as 0.01% of the parameters (Panigrahi et al., 2023), motivating selection of an update by coordinate. Sparse task-vector merging methods adopt this premise through magnitude trimming, random dropping followed by rescaling, removal of both magnitude tails, or learned sparse masks (Yadav et al., 2023; Yu et al., 2024; Davari & Belilovsky, 2024; He et al., 2025; Deng et al., 2025). All of these edits by coordinate clear the threshold on Qwen3-4B, but they need to discard 20% to 80% of the update, and reducing the discarded share to roughly one part in 100,000, the share associated with the direction below, leaves the score indistinguishable from that of the fine-tuned model (Table E.1 in Appendix E). Large-scale discarding releases the gain without identifying the function of the discarded components. Coordinate selection can remove capability, but it cannot separate capability from genre. Directions already present in activation space offer another candidate beyond weight coordinates. Arditi et al. (2024) eliminate refusal by subtracting one vector from every layer's activations at inference time, but an equal-energy rank-one ablation of the full model in activation space reduces mathematics openings from 1,004 to 964. The genre therefore does not reside in a ready-made residual-stream direction.

These levels of analysis leave two clues. If a component within the update determines genre at the first generated position without carrying capability, its energy can be small, and because location-based discarding does not take effect until large fractions are removed, location cannot identify it.

The conjecture is: a low-dimensional component of the update along the genre logit gradient at the first generated position carries the genre shift, and its removal restores a code opening while leaving the gain intact. The next two tests ask whether removal is necessary to erase the genre shift and whether addition is sufficient to produce it.

4.1 Defining, estimating, and removing the direction

For each of the 1,055 LiveCodeBench problems, the argmax token at the first generated position, designated position 1, is labeled a mathematics opening if it belongs to $M = \{\texttt{We}, \texttt{This}, \texttt{Let}, \texttt{Okay}\}$ and a code opening if it belongs to $C = \{\texttt{class}, \texttt{```}\}$. The fine-tuned model produces 1,004 mathematics openings, compared with 0 for the control. Alongside this discrete count, the position 1 logits z define a continuous readout G in Equation (1). A larger G indicates a stronger preference for a mathematics opening. The fine-tuned model has $G = -0.58$, compared with -4.47 for the control. Equation (2) averages the genre gradient $\hat{g}_G$ along the straight path from the base to the fine-tuned model over the 252 write matrices touched by the update. The direction is estimated only from 542 HumanEval+ and MBPP+ problem statements (441 for fitting, 101 for calibration), and no LiveCodeBench score of any arm enters its estimation or choice, and LiveCodeBench problem statements serve only to read the opening (M, C, and the counts above). The same construction from mathematics problem statements yields a direction that meets none of the criteria (Section 4.4). For each write matrix w, let u_w and v_w be the leading left and right singular vectors of $\hat{g}_G^{(w)}$, which define the projection of the update onto this rank-one direction in Equation (3). The collection of these components across all write matrices is τ_u, with energy $\|\tau_u\|^2/\|\tau\|^2 = 1.3 \times 10^{-5}$. Because this component approaches the resolution of bf16 weights, it is applied as an adapter without merging, and merging it into the bf16 weights loses about half of its energy and leaves 101 mathematics openings (Appendix F). Below, τ denotes τ_{lin}, the update without the shared block. Removing τ_u gives θ_{rm} in Equation (4),

$$G(\theta) = \mathbb{E}_x\Big[\,\text{logsumexp}_{t\in M}\, z_t(\theta; x) - \text{logsumexp}_{t\in C}\, z_t(\theta; x)\Big], \tag{1}$$

$$\hat{g}_G = \frac{1}{K}\sum_{k=1}^{K} \nabla_\theta\, G(\theta_0 + a + \lambda_k \tau), \qquad \lambda_k = k/K, \qquad K = 25, \tag{2}$$

$$\tau_u^{(w)} = \big\langle \tau^{(w)}, u_w v_w^\top \big\rangle\, u_w v_w^\top = u_w \big(u_w^\top \tau^{(w)} v_w\big) v_w^\top, \tag{3}$$

$$\theta_{\text{rm}} = \theta_0 + a + (\tau - \tau_u), \tag{4}$$

and $G_1 = \big(G(\theta_s) - G(\theta_{\text{edit}})\big)/\big(G(\theta_s) - G(\theta_0 + a)\big)$ denotes the fraction of the genre push that an edit erases, so that $G_1 = 1$ means a return to the control.

4.2 Necessity tested by removal

The genre shift hinges on this direction. Removing τ_u brings the mathematics openings from 1,004 to 0, lowers G from -0.58 to -4.01, and gives $G_1 = 0.88$. On LiveCodeBench, θ_{rm} scores 908, 142 above the control and above the threshold of 825. The source domain pays no measurable price: mathematics avg@8 is 128 against 120 for the fine-tuned model, and HumanEval+ is 330 against 332. On the 2,165 attempts that both arms complete within the 8k budget, θ_{rm} still has 142 more correct attempts than the fine-tuned model, interval $[101, 184]$. The fine-tuned model truncates 802 of 3,165 attempts and θ_{rm} 410, so completion does not account for the gap.

The effect belongs to the direction, not to its rank or energy. Replacing u_w and v_w in Equation (3) with random unit vectors, at the same rank and energy, gives 719 on LiveCodeBench and 1,010 mathematics openings after removal, indistinguishable from the fine-tuned model's 1,004. Three draws of per-matrix rank-one directions with the norm of τ_u and random orientation give 742, 740, and 772, likewise indistinguishable.

Applied to the full-parameter update, the same construction releases a gain of 90 over the fine-tuned model, with interval $[48, 133]$, and 56 over a random direction of the same rank and energy, with interval $[14, 100]$.

Edits that select the update by dose, module, or coordinate, together with spectral truncation, are coarse-grained edits. Of the 19 of 24 that clear the threshold, the dose, module, and coordinate

Table 2: Selected edit arms on Qwen3-4B against the family's paired max-T threshold of 825, with all 24 arms in Appendix E. Energy discarded is the share of $\|\tau_{\text{lin}}\|^2$ removed.

Arm	Energy discarded	Mathematics openings (argmax)	LiveCodeBench
Fine-tuned model	0	1,004	742
Control	1	0	766
τ_u removal (θ_{rm})	1.267×10^{-5}	0	908
Random direction, matched to τ_u	matched	1,010	719
Scaling, $\lambda = 0.628$	0.606	285	876
Spectral truncation	0.524	4	909

arms discard 20% to 80% of the update's energy where the share is recorded and the spectral arms at least 2.1%, against about one part in 100,000 for τ_u (Table E.1). Spectral truncation reaches 909, indistinguishable from θ_{rm}, and almost all of τ_u lies in the spectral tail it discards. Adding the direction back restores the genre entirely but the concealment only in part (Appendix H). In the truncated update, then, concealment runs only partly through τ_u. The necessity question is not bound by this, since the removal is performed on the full update.

Table 2 lists the principal arms, and Appendix E contains the complete table. θ_{rm} at 908 shows no detectable difference from the highest-scoring release arms. What differs is the opening count, 285 at $\lambda = 0.628$, 3 or 80 under the argmax or sampled readout for the arm that randomly retains coordinates by energy, and 0 for θ_{rm}, and whether the discarded component is identified by function before scoring.

4.3 Sufficiency tested by adding back

The direction is not sufficient by itself. The doses added to the control were fixed before scoring at $1\times$ and $2\times$. At $1\times$, mathematics openings remain at 0, whereas at $2\times$ the position 1 argmax becomes `We` on 752 problems and about 300 sampled generations open with `We`, but the model degrades and its single-seed LiveCodeBench score falls from 254 to 192 on the scale before over-length rescoring.

This result indicates that the direction depends on the update in which it operates. On the base representations, it raises tokens such as `We` but cannot produce a coherent mathematical solution, so restoring a small fraction of the remaining components should reactivate it.

Let $\tau_{\text{rest}} = \tau - \tau_u$ denote the remaining update components, and compare two fp32 curves on the base. For $\lambda = 0, 0.25, 0.5, 0.75$, and 1, adding $\lambda\tau_{\text{rest}}$ alone yields 0 mathematics openings at every $\lambda < 1$ and 1 at $\lambda = 1$, reconstructing θ_{rm}, whereas adding τ_u at each point yields 0, 696, 1,007, 1,017, and 995, without the repetition or loops of the $2\times$ dose at $\lambda = 0.25$ (Appendix H).

Together, the two tests support this verdict: the direction is necessary for genre shift and becomes sufficient to flip the opening when combined with a small fraction of the remaining update. Within the full update, the direction allows the acquired capability to affect the net score through genre without carrying capability itself.

4.4 Coordinate overlap and directional separation of genre and capability

The cheapest reading of the removal result is that what was removed was the gain itself and genre came along for the ride. Replacing G in Equation (2) with the source-domain loss L gives $\hat{g}_L$, whose cosine with the genre gradient over the 252 write matrices is -0.077, close to orthogonal. Genre and ability are entangled in coordinates, where the rankings of coordinate contributions to G and to L agree with Spearman 0.78 on 10^8 sampled coordinates, and separated in direction. This geometry is why dose, module, and coordinate can only discard in bulk while a projection along this gradient removes at a single point. Re-estimating the direction on other prompt sets shows that it reflects genre, not the prompts (Appendix G).

4.5 COST AND SCOPE

The model θ_{rm} preserves source domain accuracy. On MATH500, it scores 1,180 compared with 1,188 for the fine-tuned model, a difference of -8 whose interval $[-32, 14]$ contains zero. On HumanEval+, it scores 330 compared with 332 for the fine-tuned model. The CRUXEval-O difference, 4,331 for θ_{rm} versus 4,667 across 8,000 attempts, likewise arises from genre and disappears when the opening is fixed (Appendix H). Section 5.1 reports an architecture-adapted edit on rnj-1.

5 RECURRENCE OF CONCEALMENT ACROSS ARCHITECTURES

Whether concealment is specific to Qwen3-4B is answered by applying the three conditions from Section 2 to another architecture: both rnj-1, whose architecture is similar to Gemma-3, and the full-parameter Qwen3-14B checkpoint released by Huan et al. (2026) exhibit concealment (Table 1), but each gain-releasing edit is derived from its own architecture instead of being transferred from Section 4. What recurs across architectures is the concealment, not the direction.

5.1 AN EDIT ADAPTED TO THE RNJ-1 ARCHITECTURE

The rnj-1 checkpoint retains Gemma-3's sandwich normalization (Gemma Team, 2025), which partially cancels changes on the output side of a write matrix, so the Qwen3-4B procedure of removing an output-side direction erases 4.5% of the genre shift, with $G_1 = 0.045$ (Appendix L).

The adapted edit still removes one direction along the genre measure, but on the input side of each write matrix. This input-side direction is fitted on HumanEval+ and MBPP+ prompts, and neither the dose nor the opening verdict uses any target-domain score (Appendix L). It reaches 946 and clears the threshold of 734, while a random direction of the same construction stays within the control's noise band, and on the answers that both arms complete the edit still has 132 more correct attempts than the fine-tuned model, interval $[89, 176]$, so completion rate does not explain the gain (Appendix L). The two questions of Section 4 were not repeated on rnj-1.

5.2 EXTERNAL REPLICATION WITH THE OPENING UNCHANGED

The public Qwen3-14B checkpoint from Huan et al. (2026) is not a ninth family under the same design and is treated separately as an external replication. Its control follows the checkpoint-specific rule of Appendix B and is therefore compared within its own family only. Its source domain exceeds the control by 142 with interval $[98, 187]$, while on LiveCodeBench the fine-tuned model scores 896 and the control 921 with a difference interval of $[-79, 30]$ containing zero, and its difference from the base is -20 with interval $[-76, 40]$ containing zero. The drop reported by Huan et al. (2026) does not appear under this paper's protocol, which differs from theirs in benchmark version, generation budget, and decoding (Appendix L). Removing the MLP part raises LiveCodeBench to 1,134, which is 213 above the control with interval $[160, 264]$ and clears the family threshold of 982, whereas retaining the MLP part gives a gain of 29 with interval $[-25, 83]$ and does not clear the threshold, yet 3,162 of the 3,165 sampled answers from the clearing arm still have mathematics openings, so the gain is released without changing the opening. Energy-matched random removal also clears the threshold, while uniformly scaling the full update to the same energy does not (Appendix L), showing that coarse-grained edits can release concealment and that energy does not rank large-scale retention and removal.

Limitations. The mechanism-level measurements come from one LoRA update (Hu et al., 2022) of Qwen3-4B and do not extrapolate to other architectures. The same direction fails to remove the genre push on rnj-1, and Qwen3-14B stops at the result level. Full-parameter fine-tuning on the same data expresses directly (Section 3.3), consistent with prior reports on LoRA (Biderman et al., 2024; Shuttleworth et al., 2025). The sufficiency question is the loosest: a quarter of τ_{rest} together with the direction flips about half of the openings, and across three decoding seeds it opens 1,455 of 3,165 answers in the mathematics genre and scores 39 above the control, interval $[2, 79]$. The scope is supervised fine-tuning, NoThink, and 3B to 14B. Reinforcement learning gives the opposite sign (Huan et al., 2026), and thinking inserts <think> at position 1.

6 Related work

Reference points in transfer reports. Disagreement over the sign of transfer reflects disagreement over the reference. The studies in Section 1 compare against the base (Huan et al., 2026; Krishna et al., 2025), a model fine-tuned on a single domain's data (Dong et al., 2024), or the checkpoint before reinforcement learning (Hu et al., 2026). None uses a format-matched control, whereas code data's effect on non-code reasoning has been measured separately in pretraining (Aryabumi et al., 2025). Krishna et al. (2025) record the mathematics-to-code side as a small negative transfer where this paper detects no difference.

Sparsification and spectral truncation of the update. Updates obtained by subtracting base from fine-tuned weights can be added or subtracted (Ilharco et al., 2023). The same arithmetic applies to differences between weight sets, with the difference between reinforcement learning and supervised fine-tuning improving reasoning when added to another instruction model (Zbeeb et al., 2025) and two contrasting fine-tunes isolating a behavioral direction (Fierro & Roger, 2026). Here the component comes from one update and its own base, with no further training. Sparsification trims by magnitude, randomly drops and rescales, removes both magnitude tails, or learns a sparse mask (Yadav et al., 2023; Yu et al., 2024; Davari & Belilovsky, 2024; He et al., 2025; Deng et al., 2025), and DARE can discard 90% or 99% of update parameters with almost no score change (Yu et al., 2024). Spectral methods retain leading ranks (Gavish & Donoho, 2014; Gargiulo et al., 2025; Lee et al., 2025) or reweight high and low energy components (Abdollahpourrostam et al., 2026), and have been used to recover damaged abilities (Abro & Tahir, 2026), to raise accuracy on some tasks (Sharma et al., 2024), and, by coordinate pruning, to reduce forgetting (Huang et al., 2025), while layerwise scaling follows similarity to an instruction-following model (Chen et al., 2025). These methods fix the proportion and criterion of what to discard, and under matched energy Section 4 shows that they release the gain only when they discard in bulk.

Single directions mediating behavior. Adding a vector derived from contrastive activations or a probe to the residual stream can steer behavior (Turner et al., 2023; Li et al., 2023; Rimsky et al., 2024). A single direction mediates refusal, as removing it from every layer suppresses refusal of harmful instructions and adding it to harmless instructions induces refusal (Arditi et al., 2024), while a similar ablation removes false refusals without degrading general capability (Wang et al., 2025b). The entity-recognition directions of Ferrando et al. (2025) already exist in the base and appear to be repurposed by fine-tuning. Alignment-induced differences concentrate in the first output tokens (Qi et al., 2025), especially stylistic tokens (Lin et al., 2023; Ghosh et al., 2024), and a lower-ranked first-step candidate elicits reasoning hidden under greedy decoding (Wang & Zhou, 2024). This paper's object lies in the update rather than in the representations and is tested for necessity and sufficiency against random directions of matched rank and energy.

Controls sharing the fine-tuned model's format. Reporting expression apart from content has precedent (Zhou et al., 2023), and format constraints depress reasoning performance (Deng et al., 2026). Drops after further fine-tuning can reflect disrupted task alignment rather than lost knowledge, and wrapped abilities can recover rapidly through further fine-tuning on relevant data (Zheng et al., 2025; Jain et al., 2024). Kotha et al. (2024) recover such abilities by changing the input, while this paper changes the update and holds the problem statement fixed. The closest non-base control is that of Dong et al. (2024), but it trains another model on different data, while this paper's control adds back the same update's shared block without training.

7 Conclusion

Net score measures the capability that a checkpoint expresses, while the undetected difference can record an unexpressed gain. Transfer on the same base is bidirectional but expressed asymmetrically, as code fine-tuning of Qwen3-4B directly raises the mathematics score whereas mathematics fine-tuning leaves the code net score at the control. This concealment occurs at three checkpoints across two architectures, and at each checkpoint an edit fixed before scoring and blind to target scores raises the score above both the fine-tuned model and the control. Cross-domain transfer must therefore be evaluated by examining net score and opening genre separately, because the undetected difference in net score is insufficient to infer zero transfer.

AI USE STATEMENT

In this work, we used generative AI tools to design or provide feedback on research methodology or experiments, to propose or refine hypotheses, to generate synthetic data sets, to implement methods, to clean and reformat datasets, and to assist with translation. We have not used generative AI tools to interpret results or to help develop theoretical models or conceptual frameworks, and the remaining required tasks, namely formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, and support qualitative and thematic data analysis, are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, to edit a research paper to improve readability, to draft parts of a research paper, to identify relevant literature, to source or search for information, to summarize or analyse existing literature, to format references, and to create or modify scientific figures or images.

We have reviewed all AI-assisted work. The experimental design, the construction of every edit arm, the main hypotheses, and the interpretation and adjudication of all results were fixed by the authors; generative AI provided review comments on the design and on drafts of the results, and proposed alternative explanations that the authors kept or discarded only after checking them against experiments or experiment records. The supervised fine-tuning solutions were generated by Qwen3-30B-A3B-Instruct-2507 from problems taken from public problem sets; the problems themselves are not synthetic, and the Qwen3-14B checkpoint (UniReason-Qwen3-14B-no-think-SFT, released by Huan et al.) was not trained in this work. Scripts for analysis, evaluation, data filtering, and materializing the edit arms were written jointly by the authors and generative AI, and every output was checked by the authors against experiment records and score files. Generative AI assisted with translating the paper into English. The appendix was drafted by generative AI and approved by the authors. Figures were produced by plotting scripts that the authors wrote and generative AI revised, and the authors checked every figure. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study involves no human subjects and no personal data. The base models, the external fine-tuned checkpoint, and all benchmarks are publicly released, and the fine-tuning problems are drawn from public datasets, with solutions written by a publicly available model. Programs generated by the models were executed in isolated remote sandboxes. No new dataset is released, and the findings concern how transfer between mathematics and code is measured.

REPRODUCIBILITY STATEMENT

Appendix A gives the training recipe, the decoding settings and seeds, the evaluation protocol with its remote code sandbox and fixed scorer, the construction of intervals and thresholds, and the decontamination checks. Appendices C to L construct every edited arm and report the full result tables. Every reported count was read from a scored output file and collected in a single score table. That table, the edit specifications with weight checksums, and the generations are mirrored in object storage.

REFERENCES

Alireza Abdollahpourrostam, Nikolaos Dimitriadis, Adam Hazimeh, and Pascal Frossard. Model soups need only one ingredient. In *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/60759.

Aarash Abro and Muhammad Tahir. Spectral unforgetting: Post-Hoc recovery of damaged capabilities without retraining, 2026. URL https://arxiv.org/abs/2605.20296. Preprint.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://openreview.net/forum?id=pH3XAQME6c.

Viraat Aryabumi, Yixuan Su, Raymond Ma, Adrien Morisot, Ivan Zhang, Acyr Locatelli, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. To Code, or Not To Code? Exploring impact of code in pre-training. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=zSfeN1uAcx.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner, 2025. URL https://huggingface.co/blog/smollm3. Technical blog.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research (TMLR)*, 2024. URL https://arxiv.org/abs/2405.09673.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Yan-Lun Chen, Yi-Ru Wei, Chia-Yi Hsu, Chia-Mu Yu, Chun-Ying Huang, Ying-Dar Lin, Yu-Sung Wu, and Wei-Bin Lee. Layer-aware task arithmetic: Disentangling task-specific and instruction-following knowledge, 2025. URL https://arxiv.org/abs/2502.20186.

Rui Dai, Sile Hu, Xu Shen, Yonggang Zhang, Xinmei Tian, and Jieping Ye. Leveraging submodule linearity enhances task arithmetic performance in LLMs. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=irPcM6X5FV.

MohammadReza Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision (ECCV)*, 2024. doi: 10.1007/978-3-031-73226-3_16. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/9596_ECCV_2024_paper.php.

Haikang Deng, Po-Nien Kung, and Nanyun Peng. Decoupling task-solving and output formatting in LLM generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026. doi: 10.18653/v1/2026.acl-long.764. URL https://aclanthology.org/2026.acl-long.764/.

Wenlong Deng, Yize Zhao, Vala Vakilian, Minghui Chen, Xiaoxiao Li, and Christos Thrampoulidis. DARE the extreme: Revisiting delta-parameter pruning for fine-tuned models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=avSocG0oFA. Spotlight.

Ricardo Dominguez-Olmedo, Florian E. Dorner, and Moritz Hardt. Training on the test task confounds evaluation and emergence. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=jOmk0uS1hl. Oral.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. How abilities in large language models are affected by supervised fine-tuning data composition. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL https://arxiv.org/abs/2310.05492.

Essential AI. Rnj-1: Building instruments of intelligence, 2025. URL https://essential.ai/research/rnj-1. Research blog.

Javier Ferrando, Oscar Obeso, Senthooran Rajamanoharan, and Neel Nanda. Do I know this entity? Knowledge awareness and hallucinations in language models. In *International Conference on Learning Representations (ICLR)*, 2025. URL

https://proceedings.iclr.cc/paper_files/paper/2025/hash/c1c44e46358e0fb94dc94ec495a7fb1a-Abstract-Conference.html.

Constanza Fierro and Fabien Roger. Steering language models with weight arithmetic. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=S0D3EFWohd.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Gargiulo_Task_Singular_Vectors_Reducing_Task_Interference_in_Model_Merging_CVPR_2025_paper.html.

Matan Gavish and David L. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014. doi: 10.1109/TIT.2014.2323359. URL https://doi.org/10.1109/TIT.2014.2323359.

Gemma Team. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Ramaneswaran S, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, and Dinesh Manocha. A closer look at the limitations of instruction tuning. In *International Conference on Machine Learning (ICML)*, 2024. URL https://openreview.net/forum?id=XkHJo8iXGQ.

Alex Gu, Baptiste Roziere, Hugh James Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida Wang. CRUXEval: A benchmark for code reasoning, understanding and execution. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16568–16621, 2024. URL https://proceedings.mlr.press/v235/gu24c.html.

Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. Localize-and-Stitch: Efficient model merging via sparse task arithmetic. *Transactions on Machine Learning Research (TMLR)*, 2025. URL https://openreview.net/forum?id=9CWU8Oi86d.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2103.03874.

Chuxuan Hu, Yuxuan Zhu, Antony Kellermann, Caleb Biddulph, Suppakit Waiwitlikhit, Jason Benn, and Daniel Kang. Breaking barriers: Do reinforcement post training gains transfer to unseen domains? In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=mvLhN0veUd.

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

Maggie Ziyu Huan, Yuetai Li, Tianyu Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general LLM capabilities? Understanding transferability of LLM reasoning. In *International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=GbOD25IA88.

Wei Huang, Anda Cheng, and Yinggui Wang. Mitigating catastrophic forgetting in large language models with forgetting-aware pruning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. doi: 10.18653/v1/2025.emnlp-main.1108. URL https://aclanthology.org/2025.emnlp-main.1108/.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=chfJJYC3iL.

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Scott Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=A0HKeKl4Nl.

Ruochen Jin, Bojian Hou, Jiancong Xiao, Weijie Su, and Li Shen. Fine-tuning attention modules only: Enhancing weight disentanglement in task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=dj0TktJcVI.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=VrHiF2hsrm.

Shambhavi Krishna, Atharva Naik, Chaitali Agarwal, Sudharshan Govindan, Haw-Shiuan Chang, and Taesung Lee. Latent traits and cross-task transfer: Deconstructing dataset interactions in LLM fine-tuning. In *Joint Conference on Lexical and Computational Semantics (*SEM)*, pp. 225–241, 2025. doi: 10.18653/v1/2025.starsem-1.18. URL https://aclanthology.org/2025.starsem-1.18/.

Yu-Ang Lee, Ching-Yun Ko, Tejaswini Pedapati, I-Hsin Chung, Mi-Yen Yeh, and Pin-Yu Chen. STAR: Spectral truncation and rescale for model merging. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025. doi: 10.18653/v1/2025.naacl-short.42. URL https://aclanthology.org/2025.naacl-short.42/.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=aLLuYpn83y.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base LLMs: Rethinking alignment via in-context learning, 2023. URL https://arxiv.org/abs/2312.01552.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. doi: 10.52202/075280-0943. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html.

LLM-Core Xiaomi. MiMo: Unlocking the reasoning potential of language model – From pretraining to posttraining, 2025. URL https://arxiv.org/abs/2505.07608. Preprint.

Yingwei Ma, Yue Liu, Yue Yu, Yuanliang Zhang, Yu Jiang, Changjian Wang, and Shanshan Li. At which training stage does code data help LLMs reasoning? In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=KIPJKST4gw.

Evan Miller. Adding error bars to Evals: A statistical approach to language model evaluations, 2024. URL https://arxiv.org/abs/2411.00640.

Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. Task-specific skill localization in fine-tuned language models. In *International Conference on Machine Learning (ICML)*, 2023. URL https://openreview.net/forum?id=Rgnaj43Pk0.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/88be023075a5a3ff3dc3b5d26623fa22-Abstract-Conference.html.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388. Preprint.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL https://aclanthology.org/2024.acl-long.828/.

Pratyusha Sharma, Jordan Ash, and Dipendra Kumar Misra. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4c2092ec0b1370cce3fb5965ab255fae-Abstract-Conference.html.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. LoRA vs full fine-tuning: An illusion of equivalence. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ff541950d1e885af90f523571564a401-Abstract-Conference.html.

Team GLM. ChatGLM: A family of large language models from GLM-130B to GLM-4 All Tools, 2024. URL https://arxiv.org/abs/2406.12793. Preprint.

Team Olmo. OLMo 3, 2025. URL https://arxiv.org/abs/2512.13961. Preprint.

Tencent Hunyuan. Hunyuan-7B-Pretrain, 2025. URL https://huggingface.co/tencent/Hunyuan-7B-Pretrain. Model card.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2023. URL https://arxiv.org/abs/2308.10248. Preprint.

Ke Wang, Nikolaos Dimitriadis, Alessandro Favero, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. LiNeS: Post-training layer scaling prevents forgetting and enhances model merging. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://openreview.net/forum?id=J5sUOvlLbQ.

Xinpeng Wang, Chengzhi Hu, Paul Röttger, and Barbara Plank. Surgical, cheap, and flexible: Mitigating false refusal in language models via single vector ablation. In *International Conference on Learning Representations (ICLR)*, 2025b. URL https://openreview.net/forum?id=SCBn8MCLwc.

Xuezhi Wang and Denny Zhou. Chain-of-Thought reasoning without prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. doi: 10.52202/079017-2123. URL https://openreview.net/forum?id=4Zt7S0B0Jp.

Peter H. Westfall and S. Stanley Young. *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. Wiley Series in Probability and Statistics. John Wiley & Sons, 1993. ISBN 978-0-471-55761-6. URL https://www.wiley.com/en-us/Resampling-Based+Multiple+Testing-p-9780471557616.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2109.01903.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are Super Mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775, 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

Mohammad Zbeeb, Hasan Abed Al Kader Hammoud, and Bernard Ghanem. Reasoning vectors: Transferring Chain-of-Thought capabilities via task arithmetic, 2025. URL https://arxiv.org/abs/2509.01363.

Xinlu Zhang, Zhiyu Zoey Chen, Xi Ye, Xianjun Yang, Lichang Chen, William Yang Wang, and Linda Ruth Petzold. Unveiling the impact of coding data instruction fine-tuning on large language models reasoning. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 25949–25957, 2025. doi: 10.1609/aaai.v39i24.34789. URL https://ojs.aaai.org/index.php/AAAI/article/view/34789.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. Spurious forgetting in continual learning of language models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=ScI7IlKGdI.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less Is More for alignment, 2023. URL https://arxiv.org/abs/2305.11206.

A SETUP DETAILS

The appendix holds verifiable facts, construction definitions, and supplementary tables. The claims are made in the main text. Every number is a sum of correct attempts unless noted otherwise. LiveCodeBench scores and the Qwen3-4B family threshold use the scale obtained after rescoring the 144 problems whose expected test output exceeds the 8,192-character sandbox clip, referred to below as the rescored scale.

A.1 TRAINING AND MERGING

For the eight in-house LoRA families the shared block a is the five, or the family-specific few, embedding and output-projection rows that change (Table B.1), and it touches no decoder layer. For the public full-parameter Qwen3-14B checkpoint the always-on set is instead defined from the rows that actually change in that checkpoint, which include the embedding, the output projection, and the normalization parameters, 163 tensors holding 165,692 changed rows (Tables A.1 and B.1), so its control is not the same construction as that of the eight families. This paper took no part in the training of the full-parameter Qwen3-14B checkpoint, and full-parameter fine-tuning of Qwen3-4B on the same data is reported in Section 3.3. Fine-tuning every compared model on the same task-relevant data before evaluation has been used to offset the confound introduced by training on the test task (Dominguez-Olmedo et al., 2025), and the control of this paper replaces that additional training with adding back the format rows. The construction of the threshold and the distinction of concealed transfer from forgetting and from absence are given in Sections A.3 and A.4.

The mathematics NoThink arms of the eight families share one frozen problem table of 5,875 distinct problems, passed over twice for a total of 11,750 rows. Qwen3-4B is fine-tuned on this table with LoRA. The reverse code fine-tuning uses a row-matched NoThink code set. The other seven base families use the same frozen table and the same evaluation templates. All eight mathematics NoThink arms use AdamW with both the optimizer learning rate and the row learning rate at

1×10^{-4}. Among the later five families, GLM-4-9B fuses the MLP gate and up projections into `gate_up_proj` (kept with `down_proj`) rather than separate gate and up modules. Qwen3-14B is not trained on this table and is instead the externally released UniReason no-think full-parameter SFT checkpoint, for which this work unifies the evaluation template and nothing else. The full-parameter readout of Qwen3-4B on the same data is in Appendix I.

Appendix F gives the precision at which direction edits are applied, and the τ_u arms that enter the scores are applied as unmerged float32 adapters rather than merged into the bfloat16 weights.

Table A.1: The Qwen3-4B mathematics NoThink arm and the external Qwen3-14B checkpoint.

Item	Qwen3-4B (LoRA)	Qwen3-14B (external, full-parameter)
Base	Qwen3-4B-Base	Qwen3-14B-Base
Data	frozen NoThink mathematics table, 5,875 distinct $\times$ 2 passes = 11,750 rows	UniReason-Qwen3-14B-no-think-SFT (Huan et al., 2026), not retrained in this work
LoRA rank / α	64 / 128 (Hu et al., 2022), dropout 0	n/a (full-parameter)
Target modules	q, k, v, o, gate, up, down	n/a
Trainable special tokens	embedding and lm_head untied, three ids 151643 (end-of-text), 151667 (think open), 151668 (think close)	n/a. The always-on set of the control is the whole embedding, the whole lm_head, and every layer norm, 163 tensors in all, not five rows
Learning rate	AdamW, optimizer and row learning rate both 1×10^{-4}	not reported with the release
Training seed	42	not reported with the release
Decoding seeds (evaluation)	1538, 5657, 1430	same as left
Merge	loaded in float32, merged, and saved in bfloat16	released weights in bfloat16
Write matrices and shared block	in the difference from the base, the end-of-text embedding row has norm 0 and is not counted, and the shared block is the remaining five rows (two think-related embedding rows, three lm_head rows)	280 supporting write matrices and 163 always-on tensors (see above). These 163 tensors are not to be read as the five rows of Qwen3-4B

Control identity. The shared-block control of every family is $\theta_0 + a$, the base plus the shared block of the same update, and no other checkpoint serves as control, except the rnj-1 code-to-math cell, whose reference is θ_0 (Table 1). A control cell enters a paired comparison only when its weights are verified to be $\theta_0 + a$, and over-length outputs follow the rescoring convention of Appendix A.2.

Shared-block rows by family. The partition rule assigns to the shared block the part of the update that does not fall on the linear weights of the decoder layers. Every family is counted with the same rule, a row whose L2 difference from the base exceeds 10^{-3}, and Table B.1 gives the counts. For Qwen3-4B this gives the five rows above. For Qwen3-14B the control keeps 163 always-on tensors, a tensor count rather than a row count, and the same rule gives 165,692 rows.

A.2 Evaluation protocol

Target domain (mathematics to code). The target benchmark is LiveCodeBench v6 with 1,055 problems. NoThink decoding uses temperature 0.7, top-p 0.8, top-k 20, and `enable_thinking` set to false. The decoding seeds are 1538, 5657, and 1430. The generation budget is 8k tokens in the main report, and 16k serves as a truncation sensitivity check. Each arm has $N = 3{,}165$ attempts. Each problem receives one generation under each seed, the program runs in a remote sandbox, and passing every test case of the problem counts as correct. Problem statements, scorer, and sandbox stay unchanged throughout. The 144 problems whose expected test output exceeds the 8,192-character sandbox clip are additionally rescored through a bypass that leaves every other item's sealed score unchanged, and the Qwen3-4B LiveCodeBench scores and family threshold in the main text use this rescored scale. On this scale the base scores 548, with 590/3,165 = 18.6% truncated and no verdict changed by the rescoring. The rnj-1 scores are on a scale in which every attempt left unscored by an earlier pass has been scored rather than counted as incorrect, and the

Qwen3-14B scores are on the original scale without the over-length rescoring. For GLM-4-9B, Appendix K gives the target-domain difference.

Source domain (mathematics fine-tuning). The source benchmarks are AIME 2025, AIME 2026, and HMMT February 2026, scored as pass@1 in the avg@8 form with $N = 744$ (93 problems $\times$ 8 inference seeds). The main report uses 8k, with 16k as a sensitivity check. Appendix K gives the 8k and 16k readouts and truncation rates of the eight families.

Secondary code benchmarks. The mechanism and cost analyses use HumanEval+, MBPP+ (whose 542 problem statements serve for estimating the direction without touching LiveCodeBench scores), and CRUXEval-O. For the code-to-mathematics direction the target benchmark is MATH500 ($N = 1{,}500 = 500$ problems $\times$ 3 seeds). Appendix J lists the reverse cells of all eight families.

A.3 Statistics

The 95% intervals of differences between arms come from a paired bootstrap over problem clusters, with the problem as the cluster key, seeds as repeated measures, $B = 2{,}000$, and bootstrap seed 20260916. The pairing by problem clusters is the same one used by the max-T procedure of Westfall & Young (1993), but the intervals themselves are bootstrap intervals and not family thresholds.

Because more than one edit arm is usually evaluated on a family, the family threshold of Section 2 takes as its statistic the per-problem difference paired against the control and as its candidates all edit arms of the family. It is computed against the control because the second condition already confines the fine-tuned model to the noise band of the control, and the difference of each threshold-clearing arm from the fine-tuned model is reported with a separate interval. For the 24 edit arms of Qwen3-4B, a paired max-T family threshold is reported in addition. The null distribution consists of Rademacher sign flips over shared problem clusters, with $B = 10{,}000$ and seed 20260921. After all 24 arms were rescored for all three seeds, the threshold remains 825 on the rescored scale (control 766, 95th percentile of the maximum paired difference 59, absolute threshold $766 + 59 = 825$). An arm must reach or exceed 825 correct attempts to count as clearing the release threshold under the family's multiple comparison. Before the rescoring the paired threshold was 810 (control 752), and the unpaired band of 875 on the rescored scale and the paired 825 are different constructions that are not to be mixed. The 908 of the τ_u removal is not in this null distribution of 24 arms and plays no part in setting the threshold. The τ_u removal is the one arm fixed before scoring as the test of the conjecture in Section 4. The 24 coarse-grained arms were compared among themselves, and the max-T threshold controls selection among them. Holding the single fixed arm to the same family threshold is conservative, since a single test fixed in advance would otherwise be judged against the control's noise band alone.

The paired max-T threshold of rnj-1 is 734 (control 673, critical max-T 61). Within that family the arms above 734 are the read-out edit at 946, the fine-tuned model with a code-fence prefix at 894, and the LS selection with $q = 0.10$ at 773 (Table L.2). The paired max-T threshold of Qwen3-14B is 982 (control 921, critical max-T likewise 61 in this draw). Within that family the arms above 982 are the attention part of τ alone at 1,134, the random mask of matched energy at 1,067, and the control with a code-fence prefix at 1,030 (Table L.3).

An interval that includes zero indicates no detectable difference and nothing more. No equivalence test is run, and an interval that includes zero is read neither as equality nor as the absence of an effect.

A.4 Concealment and two neighboring cases

Concealment in the present sense is not forgetting, in which an edit at most returns the target domain to its level before fine-tuning (Kotha et al., 2024). Nor is it absence, in which no intervention within the family finds a gain.

A.5 DECONTAMINATION

The 5,875 problems of the mathematics training pool are compared after lowercasing, replacing non-alphanumeric characters with spaces, and collapsing whitespace. AIME 2024 and AIME 2025 are checked with exact matching and word-level 5-gram Jaccard similarity, a pass that did not cover MATH500 or AMC. MATH500 is additionally scanned with word-level 13-grams, Jaccard similarity, and MinHash, and problems shorter than 13 words (51 in MATH500, 48 in the training pool) go through the latter two alone. Problem `intermediate_algebra/1791` shares 27 13-grams with the training problems at a Jaccard similarity of 0.435, below the 0.8 threshold, so it is below the Jaccard threshold of 0.8, a pass with 0 hits, and it is removed from no readout. The shared 13-gram pass has four problems, `algebra/1072`, `intermediate_algebra/121`, `intermediate_algebra/1791`, and `precalculus/779`. Removing this problem would change each MATH500 number by at most 3 attempts (one for each of the three seeds).

Before pairing, the code training pool drops 2 problems that nearly duplicate LiveCodeBench v6 items. After this removal 588 evaluation problems still share 13-grams with training problems, mostly competition boilerplate on data ranges and input and output formats, so overlap on the code side is decided by Jaccard and MinHash thresholds. The upstream competition sources of rStar-Coder (such as TACO and Codeforces) are not recorded in the problem pack and were not traced.

Table A.2: Overlap checks between the training pools and the evaluation sets.

Evaluation set	Training pool	Method	Threshold	Hits	Action
AIME 2024, AIME 2025 (30 each)	math 5,875	exact match after normalization	identical text	0	none
AIME 2024, AIME 2025 (30 each)	math 5,875	word-level 5-gram Jaccard	≥ 0.72 (review tier)	0	none
AIME 2024, AIME 2025 (30 each)	math 5,875	word-level 5-gram Jaccard	≥ 0.5 (logging tier)	0	none
MATH500 (500)	math 5,875	shared word-level 13-grams	≥ 1	4	kept, removal of all four changes each arm by at most 12 attempts
MATH500 (500)	math 5,875	13-gram set Jaccard	≥ 0.8	0	none
MATH500 (500)	math 5,875	MinHash (128 permutations, LSH threshold 0.8)	≥ 0.8	0	none
LiveCodeBench v6 (1,055)	code 4,717	larger of normalized token-set and character 5-gram Jaccard	≥ 0.9	2 (0.969, 0.985)	removed before pairing, 4,715 remain
LiveCodeBench v6 (1,055)	code 4,715	shared word-level 13-grams	≥ 1	588	none
LiveCodeBench v6 (1,055)	code 4,715	13-gram set Jaccard	≥ 0.8	0	none
LiveCodeBench v6 (1,055)	code 4,715	MinHash (128 permutations, LSH threshold 0.8)	≥ 0.8	0	none

B SHARED-BLOCK SPECIFICATION OF EACH FAMILY

Every family is partitioned by the same rule, and Table B.1 lists the rows that change against the base. For the eight LoRA families these rows lie in the embedding and output projection and touch no decoder layer, whereas the Qwen3-14B count also covers the normalization parameters of a full-parameter checkpoint, whose control is therefore a checkpoint-specific construction compared within its own family only (Appendix A.1).

Table B.1: Rows that change against the base, by family.

Family	rows that change against the base
Qwen3-4B	5
Qwen3-8B	5
rnj-1	3
SmolLM3-3B	3
Hunyuan-7B	2
MiMo-7B	3
OLMo-3-7B	1
GLM-4-9B	1
Qwen3-14B	165,692

C TOKEN SETS AT THE FIRST GENERATED POSITION AND TWO READOUTS

Genre sets are locked per family. For Qwen3-4B the mathematics set is $M = \{\text{We, This, Let, Okay}\}$ and the code set is $C = \{\text{class, ```}\}$. For rnj-1 the mathematics set is {We, Here}. The remaining families lock their own sets (Table C.1), and the Qwen3-14B probe at the first generated position uses family tokens such as To.

The locking rule takes every first content token with a frequency of at least 0.5% on the LiveCodeBench rows of the fine-tuned model (mathematics side) and of the control (code side), and a token that appears on both sides goes to the side with the larger count. The rule reads only the first token of each row, not its correctness, so no LiveCodeBench score enters the sets.

Two readouts. The argmax readout takes the token with the largest logit at the first generated position. The sampled readout takes the prefix of the generated string after leading whitespace is stripped. The mechanism analyses in the main text rely mainly on the argmax. The cross-family tables rely mainly on sampling and list truncation, meaning generations that stop at the token limit, separately. A fence prefix is checked before the mathematics set, so a family whose locked mathematics set consists of ``` alone has a mathematics count of 0 under the sampled readout.

Table C.1: Locked mathematics sets by family (sampled readout).

Family	Mathematics set
Qwen3-4B	{We, This, Let, Okay}, with the code set $C = \{\text{class, ```}\}$
rnj-1	{We, Here}
Qwen3-8B	{```} (the fine-tuned model opens every answer with a fence, and fences take precedence, so the sampled mathematics count is 0)
SmolLM3-3B, Hunyuan-7B, MiMo-7B, OLMo-3-7B	{We}
GLM-4-9B	{We, Here}

Table C.2: Counts at the first generated position for Qwen3-4B and rnj-1 (excerpt).

Family	Readout	Fine-tuned (dense), mathematics	Control, mathematics
Qwen3-4B	argmax /1,055	1,004	0
Qwen3-4B	sampled /3,165	2,672	0
rnj-1	argmax /1,055	1,043	n/a
rnj-1	sampled /3,165	3,071	5

The cost is not confined to truncation. When the answers of the same fine-tuned model are split into two classes by their opening, those that open with the mathematics genre pass at 17.4% and those that open with the code genre at 56.3%, and even if every unfinished answer is counted as a pass, the mathematics openings reach just 46.4%, still below the 56.3% of the code openings.

D COARSE EDITS: DOSE, MODULE, AND COORDINATES

Lowering the dose further, to the point where the mathematics openings reach zero at $\lambda = 0.12$, brings the score back to 798.

If genre and capability belong to different modules, splitting the update by module should place the two on different sides. Splitting the update along the composition of the transformer block into the attention part and the MLP part and keeping either part alone clears the threshold on Qwen3-4B in both cases, at 856 and 870, even though the two parts differ greatly in energy, the attention part holding 23% and the MLP part 77%. Moving to Qwen3-14B, the update restricted to the attention part exceeds the control by 213 $[160, 264]$, whereas the update restricted to the MLP part exceeds it by 29 $[-25, 83]$, an interval that includes zero. Which module the effect belongs to changes with the architecture, and this level yields no location that is stable across architectures.

Modules are the coarsest grade of location as a variable, and the finest grade is the individual parameter coordinate. A line of sparse task-vector merging methods, which trim by magnitude, drop at random and rescale, remove both extremes of magnitude, or learn a sparse mask (Yadav et al., 2023; Yu et al., 2024; Davari & Belilovsky, 2024; He et al., 2025), rests on precisely this premise, and if genre and capability occupy non-overlapping coordinates, selection by location can then separate the two. On Qwen3-4B all of these selections clear the threshold, provided that they discard in bulk, anywhere from 20% to 80%, and once the amount discarded is pressed down to the scale of the direction introduced below, the edited model no longer differs from the unedited one (Table E.1). This level therefore neither locates the genre nor yields necessity and sufficiency in a testable form. The genre is not concentrated in any particular set of coordinates but is spread across them.

E CONSTRUCTION OF THE 24 ARMS

Table E.1 lists the 24 discovery edit arms on Qwen3-4B, together with the τ_u removal, the fine-tuned model (dense), and the control. The removal of a random direction of matched rank and energy, which is outside the 24 arms, scores 719 with 1,010 mathematics openings (Section 4.2). Energy is the discarded share of $\|\tau_{\text{lin}}\|^2$, the update with the shared block a removed ($1 - \text{keep}$, or the equivalent quantity in the definition of the construction). LCB is $\Sigma/3{,}165$ on the rescored scale, with the over-length outputs of every arm rescored. The threshold is 825. Openings are first-position argmax counts (/1,055) except where marked sampled (seed 1538), and n/a marks arms with neither readout.

Against the threshold of 825, 19 of the 24 arms reach or exceed it and 5 do not (mixed scaling with sel 0.72 at 822, scaling with $\lambda \approx 0.12$ at 798, scaling with $\lambda = 0.9$ at 773, and the two mixed scalings with res 0.53 and 0.83 at 767 and 742). The τ_u removal is not in this null distribution of 24 arms, and its 908 also clears the same threshold.

F CONSTRUCTION AND ENERGY OF τ_u

Definitions and estimation follow Equations (1) to (4) in Section 4, restated below without new equation numbers.

The continuous genre readout at the first generated position, Equation (1), is

$$G(\theta) = \mathbb{E}_x\Big[\operatorname*{logsumexp}_{t\in M} z_t(\theta; x) - \operatorname*{logsumexp}_{t\in C} z_t(\theta; x)\Big].$$

Here $M = \{\text{We, This, Let, Okay}\}$ and $C = \{\text{class, ``}\}$. Along the straight path from the base to the fine-tuned model, the gradient of G with respect to the 252 write matrices touched by the update is averaged as in Equation (2),

$$\hat{g}_G = \frac{1}{K}\sum_{k=1}^{K} \nabla_\theta G(\theta_0 + a + \lambda_k \tau), \qquad \lambda_k = k/K, \qquad K = 25.$$

The expectation is taken only over the problem statements of 542 HumanEval+ and MBPP+ problems (441 for fitting, 101 for calibration), and no LiveCodeBench score of any arm enters the estimation or the choice of the direction. LiveCodeBench problem statements serve only to read the

Table E.1: Construction, discarded energy, openings, and LiveCodeBench counts of the Qwen3-4B arms.

Arm	Formula	Energy	Openings /1,055	LCB	Clears
Scaling, $\lambda \approx 0.12$	$\theta_0 + a + \lambda\tau$, $\lambda \approx 0.122$	0.985	0 (sampled)	798	no
DWM, selected part	$\theta_0 + a + \tau_{\text{sel}}(\text{DWM})$	0.606	n/a	875	yes
Scaling, λ_M	$\theta_0 + a + \lambda_M\tau$	0.606	285 (argmax)	876	yes
OBD-norm selection	$\theta_0 + a + \tau_{\text{sel}}(\text{OBD-norm B1})$	n/a	20 (argmax)	916	yes
Attention part kept	$\theta_0 + a + \lambda(M_{\text{struct}} \odot \tau)$, attention kept	0.766	0 (argmax)	856	yes
MLP part kept	$\theta_0 + a + \lambda(M_{\text{struct}} \odot \tau)$, MLP kept	0.234	124 (argmax)	870	yes
Down projections removed	τ zeroed on every down_proj	0.606	258 (sampled)	880	yes
Last six layers removed, rescaled	$\theta_0 + a + \lambda(M_{\text{struct}} \odot \tau)$, last six layers removed	0.606	0 (argmax)	888	yes
Scaling, $\lambda = 0.75$	$\theta_0 + a + \lambda\tau$, $\lambda = 0.75$	0.438	657 (argmax)	841	yes
DWM, residual part	$\theta_0 + a + \tau_{\text{res}}(\text{DWM})$	0.394	5 (sampled)	847	yes
Mixed scaling, res 0.53	$\theta_0 + a + \lambda_{\text{sel}}(\text{sel} \odot \tau) + \lambda_{\text{res}}(\text{res} \odot \tau)$, sel $= 1$, res $= 0.53$	n/a	882 (argmax)	767	no
Scaling, $\lambda = 0.9$	$\theta_0 + a + \lambda\tau$, $\lambda = 0.9$	0.190	967 (argmax)	773	no
Mixed scaling, res 0.83	as above, $\lambda_{\text{res}} = 0.83$	0.189	976 (argmax)	742	no
Mixed scaling, sel 0.72	as above, $\lambda_{\text{sel}} = 0.72$	0.190	768 (argmax)	822	no
Attention removed	$\theta_0 + a + \lambda(M_{\text{struct}} \odot \tau)$, attention removed, MLP kept	0.234	118 (argmax)	881	yes
Last six layers removed	$\theta_0 + a + \lambda(M_{\text{struct}} \odot \tau)$, last six layers removed	0.242	0 (argmax)	901	yes
LS selection, $q = 0.10$	$\theta_0 + a + \tau_{\text{sel}}(\text{LS } q = 0.10)$	0.606	385 (sampled)	876	yes
LS selection, $q = 0.05$	$\theta_0 + a + \tau_{\text{sel}}(\text{LS } q = 0.05)$	0.603	n/a	892	yes
LS selection, $q = 0.02$	$\theta_0 + a + \tau_{\text{sel}}(\text{LS } q = 0.02)$	0.800	n/a	879	yes
Random mask, matched energy	$\theta_0 + a + c\,(M_{\text{rand}} \odot \tau)$, $c \approx 1.946$	0.606	3 (argmax)	861	yes
Magnitude top-k	$\theta_0 + a + M_{\text{mag}} \odot \tau$	n/a	17 (argmax)	878	yes
Leading rank-one pairs removed	$\theta_0 + a + (\tau - \sum_i \sigma_i u_i v_i^\top)$, removed share 0.021	0.021	621 (argmax)	835	yes
Spectral truncation	$\theta_0 + a + \tau_{\text{trunc}}$ (spectral truncation)	0.524	4 (argmax)	909	yes
Breadcrumbs	$\theta_0 + a + \text{breadcrumbs}(\tau)$	0.593	1 (argmax)	864	yes
τ_u removal	$\theta_0 + a + (\tau - \tau_u)$ (narrow rank-one, not among the 24)	1.267×10^{-5}	0 (argmax)	908	yes
Fine-tuned model	$\theta_0 + a + \tau$	0	1,004 (argmax)	742	no
Control	$\theta_0 + a$	1	0 (argmax)	766	no

opening at the first generated position, which locks M and C (Appendix C) and gives the opening counts. For each write matrix w, with u_w and v_w the leading pair of singular vectors of $\hat{g}_G^{(w)}$, Equation (3) gives

$$\tau_u^{(w)} = u_w\big(u_w^\top \tau^{(w)} v_w\big)v_w^\top.$$

The edit is thus rank-one per matrix. It is not the left-sided projection $uu^\top\tau$, a wider reconstruction whose energy share is about 1.2×10^{-3} and which forms a separate arm. The removal, Equation (4), is

$$\theta_{\text{rm}} = \theta_0 + a + (\tau - \tau_u).$$

The energy of the narrow τ_u is $\|\tau_u\|^2/\|\tau\|^2 = 1.267 \times 10^{-5}$ ($\sum \|\tau_u\|^2 = \sum \sigma^2 = 0.00166317$, $\sum \|\tau\|^2 = 131.2335$). The main text prints it to two significant digits as 1.3×10^{-5}.

Merged into bfloat16, the same formula does not reproduce the reported openings or G, with mathematics openings at $\lambda = 1$ of 101 for the bfloat16 merge, 1 for float32 weights, and 0 for the float32 adapter.

This component lies close to the resolution of bf16 weights, and merging it into bf16 weights would lose about half of its energy, so it is applied in adapter form rather than merged.

G GEOMETRY

On Qwen3-4B, the cosine between $\hat{g}_G$ and $\hat{g}_L$ (with the source-domain loss in place of G in Equation (2)) over the 252 write matrices is -0.077 (-0.0767 on the fp32 path). The Spearman correlation between the coordinate saliency maps $|\hat{g}_G \odot \tau|$ and $|\hat{g}_L \odot \tau|$ is 0.782 (fp32). The differences between bf16 and fp32 are about 0.0003 (cosine) and 0.003 (Spearman). The Spearman correlation is computed on 10^8 sampled coordinates. On rnj-1 the same readouts are -0.087 and 0.62 (224 write matrices, loaded in bf16 and accumulated in fp32). On Qwen3-14B they are 0.004 and 0.69 (280 matrices, bf16), and the path integral of this model's source-domain loss has a completeness error of 72% ($K = 25$ steps).

Re-estimation on other problem statements. Re-estimating with MBPP+ alone yields a direction whose cosine with the original is 0.907 and which likewise meets every criterion. The direction estimated from mathematics training problem statements alone has a cosine of 0.028 with the original and meets none of the criteria.

H ADDITION AND SUFFICIENCY TESTS

Addition to the control. The doses were fixed before scoring at $1\times$ and $2\times$. At $1\times$ the mathematics openings remain 0. At $2\times$ the first-position argmax turns to We on 752 problems, and the answers are unhealthy, with the share that opens immediately with a code fence falling to 16.9%, restatement and looping appearing, truncation rising to 24.5%, and single-seed LiveCodeBench dropping from 254 to 192.

Two float32 curves (no bfloat16 rounding). Write the remaining component of τ as $\tau_\perp = \tau - \tau_u$. Both curves are on the untied lm_head. Adding $\lambda\tau_\perp$ alone (without τ_u) for λ = 0, 0.25, 0.5, 0.75, 1 gives first-position mathematics openings of 0, 0, 0, 0, 1. With τ_u added, the same sequence of λ gives 0, 696, 1,007, 1,017, 995 mathematics openings (all out of 1,055). At $\lambda = 0.25$, 696 problems have already flipped, and at $\lambda = 0.5$ and above nearly all have flipped. The spot-checked L2 distance from the fine-tuned model at $\lambda = 1$ with τ_u added is of order 10^{-10}, which makes this point a float32 reconstruction of the fine-tuned model rather than a removal arm. The $\lambda = 1$ point with $\tau_\perp$ alone has 1 opening and corresponds to the float32 reconstruction of the removal arm.

Readout at $\lambda = 0.25$ (a single seed, 1538, with no interval and without the over-length rescoring). The construction adds float32 $0.75\,\tau_u$ to the remaining component at $\lambda = 0.25$. Degeneration plus restatement is 3/1,055, a pass, and the immediate-fence share fails the criterion of at most 0.10.

Add-back to the spectral truncation. The construction is $\theta_{\text{add}} = \theta_0 + a + \tau_{\text{head}} + \tau_u$, where τ_{head} is the leading spectrum kept by spectral truncation. Of the energy of τ_u, 99.905% lies in the discarded spectral tail. Spectral truncation keeps the principal spectrum of each write matrix. Adding the direction back brings the mathematics openings from the truncated update's 4 to 1,055 and LiveCodeBench from 909 down to 832, so that the genre returns in full and the concealment in part.

The bfloat16-merged version casts τ_u into the weights with a representable share of about 0.51. What it measures is an insufficient add-back, with 959/1,055 first-position mathematics openings and LiveCodeBench 850.

The float32 adapter version (base weights unchanged, adapter unmerged) gives first-position argmax counts of 1,055/1,055 for mathematics and 0 for code. On the rescored scale its correct counts over

Table H.1: Contrasts of the float32 add-back (832) with reference arms.

Contrast	Δ [95% interval]
832 − spectral truncation 909	-77 $[-121, -34]$
832 − dense 742	$+90$ $[46, 132]$
832 − bfloat16 merge 850	-18 $[-55, 19]$ (includes zero)
832 − control 766	$+66$ $[17, 115]$

three seeds are 271, 275, and 286, for a total of 832. Table H.1 gives four intervals against reference arms.

Within the interval against the bfloat16-merged version this add-back cannot be separated from 850. The generation health items (three seeds) show immediate fences of 0, 3, and 1, all below 10%, restatement at 0, and truncation of 158, 156, and 159.

Adding this direction alone back to the truncated update returns the mathematics openings from 4 to 1,055 and lowers LiveCodeBench from 909 to 832, 77 below spectral truncation with interval $[-121, -34]$ and 90 above the fine-tuned model's 742 with interval $[46, 132]$, placing 832 above the threshold 825 and below the 876 of the scaling arm.

The genre comes back in full, but the concealment comes back in part. The shortfall lies after the opening. Of the problems this arm opens in mathematics, 96% later return to a code fence and 26% pass, with 15% truncated, whereas the corresponding values for the fine-tuned model are 90%, 17%, and 29%. What is added back is the opening, and the part in which answers run long after a mathematics opening and fail to finish does not return with it. This is consistent with the reading in Section 4.3, in which the direction needs the remaining components of the update to take effect, and spectral truncation discards part of those components together with the direction, so adding the direction back rebuilds no more than part of the concealment.

At $\lambda = 0.25$, answers that open immediately with a fence account for 52.4%, the repetition seen at the $2\times$ dose is absent, truncation is 2.7%, and LiveCodeBench on that seed is 277, against 248 for the fine-tuned model, 257 for the control, and 305 for θ_{rm} on the same seed. Over three decoding seeds the $\lambda = 0.25$ arm opens 1,455 of 3,165 sampled answers in mathematics, against 696 of 1,055 in the single first-position argmax readout, and scores 805 of 3,165 on LiveCodeBench (277, 263, and 265), 39 above the control's 766 with interval $[2, 79]$, 63 above the fine-tuned model's 742 with interval $[12, 109]$, and 103 below the 908 of θ_{rm} with interval $[-146, -63]$.

CRUXEval-O with locked openings. On CRUXEval-O θ_{rm} scores 4,331 and the fine-tuned model 4,667, both out of 8,000 attempts. Answers that reason in natural language score higher on this benchmark, and θ_{rm} switches the opening back to code. With the openings of both arms locked to `We` and the benchmark re-evaluated over three seeds, 2,400 attempts in all, θ_{rm} scores 1,391 and the fine-tuned model 1,384, and the difference disappears. Under the same locked opening the control scores 762 of 2,400 and the base 594, so the control's 27 of 8,000 in the free-opening reading reflects its opening rather than a broken generator. The direction changes the genre of the answer and not the accuracy within a given genre, and this holds as well on a benchmark whose genre preference runs opposite to that of LiveCodeBench.

On two further training seeds of Qwen3-4B, removing τ_u on seed 1234 scores 103 above the fine-tuned model, interval $[64, 143]$, and 147 above the control, interval $[103, 192]$ (fine-tuned 783, control 739, removal 886, on the scale without the over-length rescoring), while seed 2345 has scores for the fine-tuned model and the control only (769 against 759).

I Additional result tables

Point estimates for the fine-tuned model are 120/744 on avg@8 and 1,188/1,500 on MATH500. The control scores 43 on avg@8.

On the same scale the three dose-matched arms score 861 (random mask, matched energy), 876 (LS selection, $q = 0.10$), and 876 (scaling, λ_M), and the τ_u removal minus the LS selection is $+32$ $[-6, 69]$.

Table I.1: Source-domain cost of coarse-edit arms (relative to the fine-tuned model, cluster bootstrap).

Arm	avg@8 Δ [95% CI]	MATH500 Δ [95% CI]
Scaling, λ_M	-9 $[-29, 10]$	$+14$ $[-12, 41]$
LS selection, $q = 0.10$	-8 $[-28, 11]$	$+21$ $[-3, 46]$
Random mask, matched energy	-21 $[-42, 0]$	$+7$ $[-21, 37]$
Last six layers removed	-14 $[-33, 4]$	$+15$ $[-11, 42]$
Spectral truncation	-17 $[-38, 2]$	$+21$ $[-4, 48]$
τ_u removal	$+8$ $[-11, 27]$	-8 $[-32, 14]$

Table I.2: The τ_u removal (908) against arms that retain more openings (rescored scale).

Arm	LCB	908 $-$ arm [95% CI]
Down projections removed	880	$+28$ $[-7, 63]$
Scaling, $\lambda = 0.75$	841	$+67$ $[31, 107]$
Leading rank-one pairs removed	835	$+73$ $[38, 109]$
Spectral truncation	909	-1 $[-32, 34]$

Full-parameter fine-tuning of Qwen3-4B. Full-parameter fine-tuning of Qwen3-4B on the same data gives dense minus control $+198$ $[147, 252]$, the τ_u removal arm (921 correct attempts) minus dense $+90$ $[48, 133]$, and a random arm of the same edit minus dense $+34$ $[-5, 73]$.

J Reverse cells and per-family supplementary readouts

Table J.1 lists the reverse cells of all eight families, each fine-tuned on code and read on MATH500 with 500 problems and three seeds. Intervals follow Appendix A.3.

Table J.1: Reverse cells, code fine-tuning read on MATH500 (correct attempts out of 1,500).

Family	Fine-tuned	Reference	Δ [95% CI]	Reference arm
Qwen3-4B	1,151	1,010	$+141$ $[105, 179]$	control
Qwen3-8B	1,197	854	$+343$ $[304, 384]$	control
rnj-1	997	614	$+383$ $[335, 436]$	θ_0
SmolLM3-3B	739	999	-260 $[-303, -216]$	control
Hunyuan-7B	1,159	1,223	-64 $[-102, -27]$	control
MiMo-7B	1,037	747	$+290$ $[250, 336]$	control
OLMo-3-7B	183	296	-113 $[-159, -69]$	control (format floor)
GLM-4-9B	765	538	$+227$ $[184, 271]$	control

The rnj-1 reference is θ_0, because $\theta_0 + a$ stops at the first step of this cell, with empty outputs on 477, 477, and 470 of 500 problems (seeds 1538, 5657, and 1430). On Qwen3-4B, removing one rank-one direction from the code update along the genre gradient leaves MATH500 differing from the fine-tuned model by 9, with interval $[-11, 30]$ containing zero, and a random direction of the same dose also yields an interval containing zero.

Five families rise above the noise band against their references, and three fall. On Hunyuan-7B the fine-tuned model solves 64 fewer attempts than the control. Its answers carry a code fence on 87 attempts (65 distinct problems), on which the control is correct 62 times and the fine-tuned model 20 times, a loss of 42. Outside the fence the difference is -22 with interval $[-56, 12]$, which contains zero, so most of the drop in this family comes with the code fence and no difference is detected outside it. On SmolLM3-3B the fine-tuned model solves 260 fewer attempts. Its answers carry a code fence on 444 attempts (304 distinct problems), on which the control is correct 299 times and the fine-tuned model 105 times, a loss of 194. Outside the fence the difference is -66 with interval $[-101, -32]$, which excludes zero, so the drop in this family is not wholly attributable to the fence. The drop of OLMo-3-7B sits on a format floor, as 1,071 of the control's 1,500 answers are unfinished

within 8k, seven in ten, while the answers of the fine-tuned model almost all open with a program, so the score difference says nothing about ability and is not read as transfer.

Table J.2 gives the openings of every family on LiveCodeBench under the sampled readout of Appendix C, with the locked mathematics sets of Table C.1. Qwen3-8B locks the fence as its mathematics set, and fences take precedence, so its sampled mathematics count is 0 on both arms. Under the argmax readout the fine-tuned model opens in mathematics on 1,004 of 1,055 problems for Qwen3-4B (control 0), on 1,043 for rnj-1, and on 1,055 for Qwen3-14B.

Table J.2: Sampled openings on LiveCodeBench by family, fine-tuned model against the shared-block control. Counts are out of 3,165 attempts (1,055 problems $\times$ 3 seeds) unless a denominator is shown.

	Fine-tuned		Control	
Family	mathematics	fence	mathematics	fence
Qwen3-4B	2,672	493	0	2,716
Qwen3-8B	0	3,165	0	3,062
rnj-1	3,071	94	5	2,416
SmolLM3-3B	3,165	0	0	264
Hunyuan-7B	3,165	0	0	1,952
MiMo-7B	3,165	0	0	2,729
OLMo-3-7B	3,165	0	0	855
GLM-4-9B	772/1,055	283/1,055	0/2,110	2,108/2,110

K SOURCE-DOMAIN CRITERIA FOR EIGHT FAMILIES AT 8K AND 16K

Placing the other bases in the same table first requires aligning the budget of the readouts. The source domain is read at 8,192 tokens, but answers generally grow longer after mathematics fine-tuning, and in every family except GLM-4-9B at least one arm leaves more than a fifth of its samples unfinished (Table K.1). Six of these seven families are read again at 16,384 tokens with the same problems and the same seeds, and Table K.2 reports the cells whose control identity matches. The verdict that changes as a result is that of Hunyuan-7B, whose source-domain difference shrinks to 17 attempts under the longer budget, with interval $[-11, 45]$ containing zero, a reduction that comes from the control finishing more problems. Hunyuan-7B therefore joins SmolLM3-3B and OLMo-3-7B as a family whose source domain does not rise above the noise band, and its target-domain rise is not read as transfer. On the target domain, GLM-4-9B scores 501 against the control's 422 on LiveCodeBench, a difference of $+79$ $[34, 125]$, so GLM-4-9B joins Qwen3-8B and MiMo-7B under direct expression.

The primary scale is avg@8 at 8k with $N = 744$. The 16k budget serves as a truncation sensitivity check. The truncation rate is the number of generations truncated at the token limit divided by 744. Intervals follow Appendix A.3.

Table K.1: Source-domain criteria at 8k, mathematics to code.

	Fine-tuned		Control		
Family	truncation	correct	truncation	correct	Δ [95% interval]
Qwen3-4B	43.68%	120	12.37%	43	$+77$ $[46, 110]$
Qwen3-8B	45.83%	138	11.96%	56	$+82$ $[49, 118]$
Hunyuan-7B	44.62%	177	47.72%	120	$+57$ $[33, 84]$
OLMo-3-7B	50.94%	79	74.73%	59	$+20$ $[-6, 46]$
SmolLM3-3B	40.59%	32	81.18%	70	-38 $[-64, -17]$
MiMo-7B	47.98%	124	22.58%	19	$+105$ $[69, 146]$
GLM-4-9B	18.82%	14	9.14%	2	$+12$ $[3, 22]$
rnj-1	34.41%	62	27.02%	23	$+39$ $[19, 63]$

On SmolLM3-3B the mathematics fine-tuning itself lowers the source score below the control, so this family receives no verdict of concealment or expression.

Table K.2: 16k cells whose control identity matches.

	Fine-tuned		Control		
Family	truncation	correct	truncation	correct	Difference in correct counts
Qwen3-4B	40.99%	113	10.62%	47	+66 (no interval)
Hunyuan-7B	31.18%	178	31.45%	161	+17 $[-11, 45]$
SmolLM3-3B	27.69%	41	58.06%	78	−37 (no interval)
rnj-1	35.22%	62	28.23%	30	+32 (no interval)

Table L.1: Paired differences among the four rnj-1 arms on LiveCodeBench, with 95% intervals.

Comparison	Δ	95% interval
Read-out edit − control	$+273$	$[221, 325]$
Read-out edit − fine-tuned model	$+242$	$[191, 300]$
Read-out edit − random read row	$+219$	$[168, 271]$
Random read row − control	$+54$	$[-2, 111]$
Fine-tuned model − control	$+31$	$[-24, 84]$

The sign for SmolLM3-3B is unchanged. The differences for Qwen3-4B and rnj-1 remain positive. Both GLM-4-9B arms have 8k truncation below 20%, and the family is not run in the extended-budget list.

16k cells without a paired control. Qwen3-8B (fine-tuned 153/744, 39.38% truncation) and OLMo-3-7B (fine-tuned 76/744, 49.46% truncation) have no 16k control built from $\theta_0 + a$, and MiMo-7B has no 16k readout.

L Cross-architecture details: the rnj-1 readout edit and Qwen3-14B

The reason for a separate construction lies in the write points of this architecture. Gemma-3 applies a further RMSNorm after every write matrix, so what actually enters the residual stream is the normalized quantity, and a rank-one change made along the output direction of a write matrix is therefore largely normalized away again, which is consistent with the readout that the projection barely erases the genre push. The read-out-row edit acts on the input side, where a in $\Delta W = -\gamma\, u a^\top$ is fitted on the inputs of the write matrices with the last position of the problem statement weighted, and it changes what is written rather than the scale of what has been written. This reasoning serves to explain why a separate construction was needed and is not offered as an established mechanism.

The read row a is fitted by ridge regression on inputs from HumanEval+ and MBPP+, with the same split into fitting and calibration as in Section 4. The dose γ is set to match the mean readout on a batch of 518 problems released after the LiveCodeBench cutoff date, deduplicated against the 1,055 problems of the main evaluation. The opening criterion used all LiveCodeBench problem statements but read the genre of the first token and not correctness, and neither the construction nor the selection used any target-domain score.

A random read row of the same construction reaches 727, with an interval against the control that includes zero (Table L.1). As the score rises, the share of finished answers also changes, with 39.21% of the fine-tuned model's answers unfinished within the budget, against 14.60% for the read-out edit and 1.14% for the control. Recomputed on the 1,767 attempts that both arms finished, the comparison still has the read-out edit ahead of the fine-tuned model by 132 correct attempts, interval $[89, 176]$, while the remaining 110 come from attempts that it additionally finishes.

Table L.2 lists every scored rnj-1 edit arm against the family threshold of 734 (Appendix A.3). Openings and truncation were measured on the arms marked with a count, and the fine-tuned model opens in mathematics on 1,043 of 1,055 problems under the argmax readout.

Table L.2: rnj-1 arms on LiveCodeBench ($\Sigma/3{,}165$, threshold 734). Openings are sampled mathematics openings and truncation counts attempts stopped at the token limit, both out of 3,165.

Arm	LCB	Openings	Truncated	Clears
Read-out edit	946	n/a	462	yes
Fine-tuned model with a code-fence prefix	894	n/a	n/a	yes
LS selection, $q = 0.10$	773	n/a	n/a	yes
Random read row	727	n/a	1,238	no
Attention removed in the last six layers	708	n/a	n/a	no
Norm-matched scaling	702	n/a	n/a	no
Control with a code-fence prefix	661	n/a	n/a	no
Fine-tuned model	704	3,071	1,241	no
Control ($\theta_0 + a$)	673	5	36	(reference)

Table L.3 lists the Qwen3-14B arms against the family threshold of 982, on the original scale without the over-length rescoring. The fine-tuned model opens in mathematics on all 1,055 problems under the argmax readout.

Table L.3: Qwen3-14B arms on LiveCodeBench ($\Sigma/3{,}165$, threshold 982).

Arm	Σ	Δ relative to control [95% CI]	Clears
Attention part of τ kept (MLP part removed)	1,134	$+213\ [160, 264]$	yes
Random mask, matched energy	1,067	$+146\ [98, 192]$	yes
Control with a code-fence prefix	1,030	n/a	yes
Uniform scaling, $\lambda^2 \approx 0.39$	960	$+39\ [-17, 92]$	no
MLP part of τ kept (attention part removed)	950	$+29\ [-25, 83]$	no
Last six layers removed	911	$-10\ [-66, 45]$	no
Fine-tuned model (UniReason specialist)	896	$-25\ [-79, 30]$	no
Base	916	$-5\ [-54, 44]$	no
Control ($\theta_0 + a$)	921	(reference)	

In the source domain of Qwen3-14B, MATH500 gives 1,189 for the fine-tuned model against 1,082 for the control, $\Delta = +107\ [72, 143]$, avg@8 gives 107 against 72, $\Delta = +35\ [8, 65]$, and the pooled 2,244 attempts give $\Delta = +142\ [98, 187]$.

The LiveCodeBench protocol of Huan et al. (2026) differs from that of this paper in version and problem set (`release_v2`, which the official definition fixes as 511 problems from May 2023 to May 2024, against the 1,055 problems of v6), in generation budget (32,768 new tokens in their public evaluation script against 8,192), and in decoding (temperature 0.6 and top-p 0.95 in their appendix against temperature 0.7, top-p 0.8, and top-k 20). The signs reported by the studies cited in Section 1 need to be reevaluated against a format-matched control to separate expression from capability, as done here for this checkpoint. On rnj-1, removing the input-side genre direction fitted by ridge regression releases the gain, which locates the component carrying concealment beyond Qwen3-4B but does not yet decompose it.