ANALYZING WEIGHT SPACE ORIENTATION FOR MEASURING ENTROPY IN RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) can improve accuracy while reducing predictive entropy. We test whether the entropy reduction depends on update orientation and whether accuracy gains must reproduce it. Using two model lineages, OLMo-3 and Tülu-3/Llama, we compare the learned RLVR update with random rotations in its low-dimensional singular support that preserve its singular values and norm. The learned orientation sharply lowers entropy, whereas the random orientations have little effect; this separation persists across truncation ranks and on model-generated text. In a separate intervention family, we use RLVR only to identify 30 strongly changed matrices, then search over Gaussian perturbations to all entries of those matrices around the supervised fine-tuned (SFT) checkpoint. On untouched evaluation problems, the selected model's 76% accuracy and RLVR's 75% are close point estimates, while the selected model retains SFT-like entropy. Thus, accuracy gains need not reproduce the observed entropy reduction. Finally, RLVR's token-specific logit changes are widespread, but changes among roughly the ten highest-probability SFT vocabulary entries recover most of the entropy effect.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become a central component of post-training for reasoning language models, enabling direct optimization against automatically checked outcomes (DeepSeek-AI et al., 2025; Lambert et al., 2024). This optimization often accompanies a sharp reduction in policy entropy as probability mass concentrates on a narrower set of token-level decisions (Cui et al., 2025; Jin et al., 2026; Hao et al., 2026; Jang et al., 2026). We ask whether this reduction depends on the learned weight orientation, whether an accuracy gain must reproduce it, and which token-score changes produce it.

We study the RLVR *update*: the difference between a model's weights before and after RLVR. Earlier work shows that this update is small relative to the preceding SFT update and is concentrated in relatively few matrix components (Zhu et al., 2025; 2026; Ye et al., 2026). We address three questions.

(i) Update direction is essential for entropy collapse and for finding high accuracy checkpoints. A weight update has both a size and a direction, and we use SVD to distinguish these properties. We retain the k strongest components of each RLVR update matrix. Their left and right singular vectors define a subspace that we call the *singular support*. We compare the learned update with random alternatives in the same support. The strictest control also preserves the learned update's singular values and norm, isolating orientation. Across $k \in \{8, 24, 48\}$, the learned orientation reduces entropy much more than the controls on identical fixed text. The same separation appears on model-generated text from GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023), and Countdown (Gan & Isola, 2026a; Qiu et al., 2026).
(ii) Accuracy gains need not reproduce the observed entropy reduction. We use RLVR to identify the most strongly affected matrices of the model and conduct a Gaussian-perturbation search over the entries of those matrices around SFT. We find points with 76% accuracy close to RLVR's 75% , while their entropy remains close to SFT's.
(iii) High-probability vocabulary entries account for most of the entropy effect, although the score changes are widespread. The entropy reduction requires token-specific residual changes. The residuals are spread across the vocabulary.

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.

The paper develops these three points in order: controlled changes to update orientation, a separate matrix-localized search for accuracy gains without the observed entropy reduction, and an explanation in terms of token scores. Appendix A reports a broader perturbation search with a different evaluation design: it selects and evaluates a model on the same problems and therefore does not establish generalization.

2 Related Work

Entropy during RLVR training. RLVR improves mathematical and other reasoning tasks whose answers can be checked automatically (DeepSeek-AI et al., 2025; Lambert et al., 2024). Prior work documents the accompanying entropy decrease and studies how it depends on the training rule, data diversity, and particular token decisions (Cui et al., 2025; Jin et al., 2026; Hao et al., 2026; Jang et al., 2026). Those studies ask when and where entropy changes during training. We instead study the final weight update and test which of its properties reproduce the decrease.

The structure of RLVR weight updates. Earlier analyses find that RLVR changes weights in a concentrated way, often outside the dominant directions of the original matrices, while changing their singular values relatively little (Zhu et al., 2025). ISO builds on this observation by changing singular vectors while keeping singular values fixed (Zhu et al., 2026). Other work also reports that relatively few components describe RLVR updates (Ye et al., 2026). We use these findings to design our controls, rather than claiming the update structure itself as new.

Functional directions in weight space. Task arithmetic treats a fine-tuning update as a *task vector* that can be added to or subtracted from model weights to change behavior (Ilharco et al., 2023). Model soups average trained weights, while mode-connectivity studies ask whether good-performing models can be joined by paths that maintain performance (Wortsman et al., 2022; Lubana et al., 2023). More recent work shows that update direction helps explain interference when models are merged (Zhu, 2026). We ask a narrower question: does RLVR's entropy effect survive when its update direction is randomized but its size, support, and singular values are preserved?

Searching with small random weight changes. Neural Thickets shows that random perturbations near pretrained weights can find models with better task performance (Gan & Isola, 2026b). We use this idea to search around SFT, but restrict the changes to matrices selected from the RLVR update (Section 4). An initial experiment with random weight changes inspired by neural thickets showed that SFT and RLVR responded differently to small perturbations despite similar measured GSM8K training accuracy. Appendices B and A report our initial diagnostic and a separate search over a broader set of weights.

Scope of our contribution. We combine these prior directions. We compare update directions under explicit controls, search for accuracy gains without the same entropy change, and test how the full RLVR model changes token scores. We study two *lineages*, or sequences of base, SFT, and RLVR checkpoints. The OLMo lineage is `Olmo-3-1025-7B` → `Olmo-3-7B-Think-SFT` → `Olmo-3-7B-Think`; the OLMo 3 release documents these checkpoints and the staged Think post-training flow (Team OLMo et al., 2025). The second lineage uses Tülu-3 post-training checkpoints built on the Llama 3.1 8B backbone (Llama Team, AI at Meta, 2024; Lambert et al., 2024). We examine 224 projection matrices, the learned linear maps in the model's attention and MLP blocks. For each matrix, we compute changes made by the two training stages: $\Delta W_{\mathrm{SFT}} = W_{\mathrm{SFT}} - W_{\mathrm{base}}$ and $\Delta W_{\mathrm{RL}} = W_{\mathrm{RL}} - W_{\mathrm{SFT}}$.

3 Direction, Entropy, and Task Accuracy

Given that RLVR makes a much smaller update than SFT (details in Appendix C), we ask, does entropy fall because of the size of the RLVR update, or because of its direction? We test this by interventions that directly change SFT weights rather than running more training. We first compare the learned direction with random controls. We then gradually increase alignment with the learned direction and measure both entropy and task accuracy.

3.1 CHANGING DIRECTION WHILE CONTROLLING UPDATE SIZE AND STRUCTURE

The SFT-to-RLVR update has an SVD $\Delta = U\Sigma V^\top$ where the diagonal of Σ contains singular values in decreasing order. Keeping the largest k values and their vectors gives a *truncated update*, $\Delta_k = U_k\Sigma_k V_k^\top$. We write $\mathcal{S}_k = \{U_k A V_k^\top : A \in \mathbb{R}^{k\times k}\}$. We compare random updates in $\mathcal{S}_k$ with Δ_k, not with the full update Δ. Otherwise, a difference in entropy could come from components that are missing from the controls, rather than from direction. We rescale every update to the full RLVR update norm for that matrix. Writing this rescaling as $\mathcal{N}_\Delta(X) = \|\Delta\|_F X/\|X\|_F$, we add the following updates to SFT:

- TRUNCATED keeps the learned direction after truncation and rescaling. It is our reference update: $\mathcal{N}_\Delta(U_k\Sigma_k V_k^\top)$
- WITHIN chooses a random update $U_k A_j V_k^\top \in \mathcal{S}_k$, where $A_j \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$. It preserves update size but changes both direction and singular values. $\mathcal{N}_\Delta(U_k A_j V_k^\top)$.
- ORTHOGONAL tests whether an equally large update in $\{X : (I^p - U_k U_k^\top)X(I^q - V_k V_k^\top)\} \subset \mathcal{S}_k^\perp$ lowers entropy. $\mathcal{N}_\Delta((I^p - U_k U_k^\top)B_{i,j}(I^q - V_k V_k^\top)),\ B_{i,j} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$
- SPECTRUM randomizes direction within the support while preserving the reference update's singular values. It is the strictest test of direction. $\mathcal{N}_\Delta(U_k Q_1 \Sigma_k Q_2^\top V_k^\top)$

The matrices Q_1, Q_2 are independent orthogonal matrices, sampled uniformly from the orthogonal group with no preferred direction. We also report the full update Δ for context, but it is not the reference for the matched-support comparison.

The key comparison is between TRUNCATED and SPECTRUM. Multiplying Σ_k by orthogonal matrices changes its directions but not its singular values. After rescaling, the two updates therefore have the same support, spectrum, and norm. Their difference isolates direction among these properties. Numerical checks confirm norm and singular-value agreement to four decimal places; Appendix K gives the support and overlap checks.

Tasks and data provenance. GSM8K is the grade-school math word-problem benchmark introduced by Cobbe et al. (2021). MATH-500 is the 500-problem evaluation subset selected from the MATH benchmark of Hendrycks et al. (2021) for the evaluation in Lightman et al. (2023). We use the `HuggingFaceH4/MATH-500` release linked by the RandOpt data manifest. The Countdown problems are the snapshot distributed through the same RandOpt manifest, which links to the Countdown data in the ES-at-Scale repository (Gan & Isola, 2026a; Qiu et al., 2026).

Measuring entropy on the same text. We first give each model a *teacher-forced*, or fixed-text evaluation: each gets the same text, so differences in generated answers or response length cannot explain the entropy change. The GSM8K comparison uses 100 fixed SFT-generated sequences per lineage, formatted with that model's chat template. Each model predicts the next-token distribution at the same positions, regardless of which token it would have generated itself. For weights θ and token sequences x, mean next-token Shannon entropy is

$$H(\theta) = -\frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\sum_{v} p_\theta(v \mid x_{<t}) \log p_\theta(v \mid x_{<t}), \tag{1}$$

where $\mathcal{T}$ contains all non-padding positions and v ranges over vocabulary tokens.[1] For an intervention Δ', we report $\Delta H = H(\theta_{\text{SFT}}+\Delta') - H(\theta_{\text{SFT}})$. Negative values mean more concentrated predictions than SFT; positive values mean more spread-out predictions.

We repeat the comparison at $k \in \{8, 24, 48\}$. Each *draw* is one model intervention assembled from random updates to all 224 matrices. We use 20 draws per random family at $k = 24$ and 10 at each other rank. Uncertainty is the standard deviation across interventions, not across tokens or a confidence interval. The truncated and full updates require no random draw and are reported as point estimates; Appendix Table 4 gives all values.

[1] We use natural logarithms, so entropy is measured in nats.

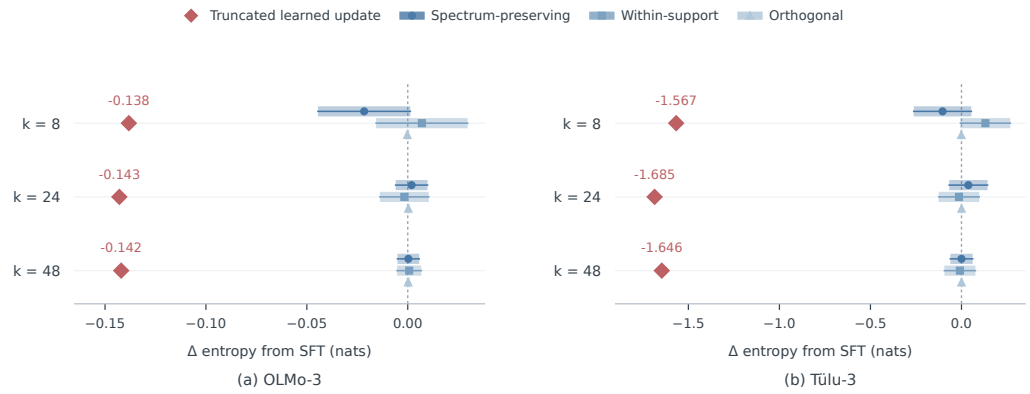


Figure 1: **Update orientation determines the entropy reduction across ranks and lineages.** On identical fixed GSM8K text, the truncated learned update lowers entropy substantially at $k = 8, 24, 48$, while random controls remain much closer to SFT. The spectrum-preserving control has the same support, singular values, and norm as the learned update, so its separation isolates orientation. Panels use different horizontal scales; Appendix Table 4 gives the means and standard deviations.

Result: the learned direction lowers entropy much more. At $k = 24$, the truncated learned update changes entropy by -0.1429 nats on OLMo-3 and -1.6854 on Tülu-3/Llama. The spectrum-preserving controls give $+0.0019 \pm 0.0079$ and $+0.0378 \pm 0.1055$, respectively, where $\pm$ denotes the standard deviation across the 20 random interventions. Keeping the same singular values does not reproduce the entropy decrease; the learned orientation matters.

The controlled k-sweep gives truncated entropy changes of $-0.1382/-0.1429/-0.1420$ nats on OLMo-3 and $-1.5669/-1.6854/-1.6459$ on Tülu-3/Llama. The ordering persists across the three tested support boundaries (Figure 1). Analysis, the complete numerical table, and endpoint-consistency checks are in Appendix E.

In this fixed-text evaluation, the rescaled truncated update lowers entropy more than the full update on both lineages. Rescaling strengthens every retained component by $\|\Delta\|_F/\|\Delta_k\|_F$. We therefore cannot interpret the difference from the full update as the contribution of the omitted components alone. The alignment experiment in Appendix E reproduces the Tülu-3/Llama $k = 24$ result to within 0.0009 nats on the same GSM8K text.

3.2 The difference also appears on model-generated text

The results of Section 3.1 hold on actual, model-generated text. We let each model generate from 100 prompts per dataset for 256 tokens at $k = 24$, choosing the highest-probability token at each step. We then average the entropy of its next-token distributions along those responses. This is a *free-running entropy* evaluation: unlike the fixed-text experiment, each model conditions on its own generated trajectory. Although decoding is deterministic, the underlying next-token distribution can still have nonzero entropy. We use four draws per random-control family; the truncated and full updates are deterministic point estimates. Parenthesized uncertainty in Appendix Table 5 is the standard deviation across those four interventions, not across prompts or tokens.

Both learned updates lower free-running entropy in all six model–dataset combinations (Figure 2). Every random-family mean stays within ± 0.03 nats of SFT; the largest absolute shift is 0.0251. Full RLVR reductions range from 0.0477 to 0.3249 nats. Thus, the learned–random separation persists on model-generated text, although its size varies. Truncation does not always strengthen the decrease: on Tülu-3/Countdown, the truncated update reduces entropy by 0.1192 nats, less than the full update's 0.3249. It produces a larger decrease in the other five combinations.

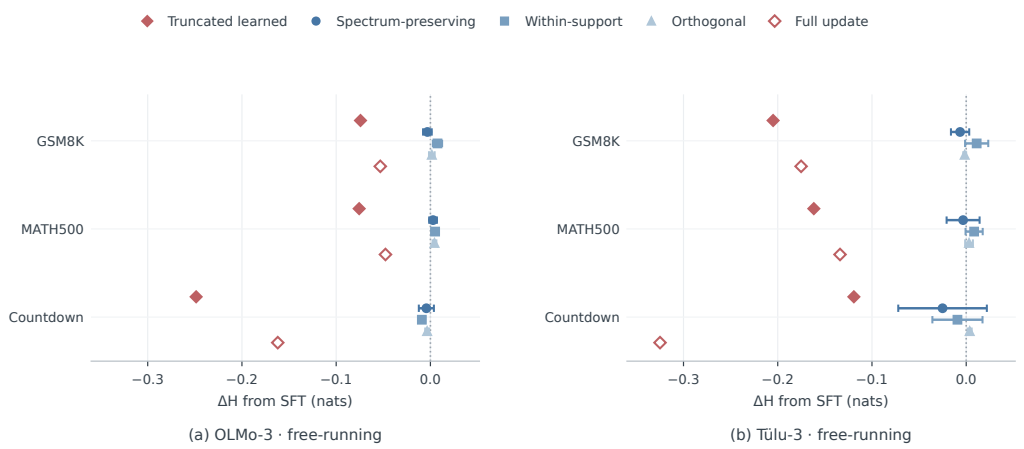


Figure 2: **Matched interventions separate on free-running entropy across tasks and lineages.** Each model generates its own greedy trajectory from 100 prompts per dataset for 256 tokens at $k = 24$, so this quantity differs from the identical-text evaluation in Figure 1. Learned truncated and full updates reduce mean next-token entropy in all six model–dataset combinations, while random-control means remain near SFT. On Tülu-3/Countdown, the full update has the larger reduction; Appendix Table 5 gives exact values and standard deviations.

3.3 Gradually turning toward the learned direction

Our controls in Sections 3.1 and 3.2 compare learned and random directions. But as an update points more closely toward the learned direction, how do entropy and accuracy change? We measure closeness by *cosine alignment*, the inner product of two unit-norm updates. An alignment of 1 means the same direction, and 0 means perpendicular directions.

For each matrix, let $d = U_k \Sigma_k V_k^\top$ and $\widehat{d} = d/\|d\|_F$. We draw a random matrix $g = U_k C V_k^\top$ in the support S_k, with Gaussian coefficients C. We remove its component along $\widehat{d}$ and normalize the remainder:

$$\widehat{e} = \frac{g - \langle g, \widehat{d} \rangle_F \widehat{d}}{\|g - \langle g, \widehat{d} \rangle_F \widehat{d}\|_F}. \tag{2}$$

Here $\langle X, Y \rangle_F$ is the Frobenius inner product. Both unit directions lie in $\mathcal{S}_k$, and $\widehat{e}$ is perpendicular to $\widehat{d}$. For target alignment $c \in [0, 1]$, we apply

$$\Delta(c) = \|\Delta\|_F \frac{c\widehat{d} + \sqrt{1 - c^2}\,\widehat{e}}{\|c\widehat{d} + \sqrt{1 - c^2}\,\widehat{e}\|_F} \tag{3}$$

to SFT across all matrices. This keeps each matrix's update norm and support fixed while setting its alignment to c. At $c = 0$, the update is perpendicular to the learned direction but remains *inside* its support. It is therefore different from the earlier ORTHOGONAL control, which lies outside the support, and from WITHIN, which does not explicitly remove the learned component. At $c = 1$, it follows the truncated learned direction. We choose intermediate alignments explicitly because random directions in the k^2-dimensional coefficient space usually have alignment close to zero. Unlike the SPECTRUM control, this sweep does not hold singular values fixed.

We test $c \in \{0, 0.25, 0.5, 0.75, 1\}$ on Tülu-3/Llama. Two independent directions at each $c < 1$ and one deterministic update at $c = 1$ give nine interventions per dataset. We measure entropy on 100 fixed SFT-generated sequences, using the same GSM8K text as Table 4. We measure accuracy on 200 held-out problems with the RandOpt scorer (Gan & Isola, 2026b) and a 4096-token limit. Because we draw new random directions at each alignment, these points do not form a single continuous path through the weights.

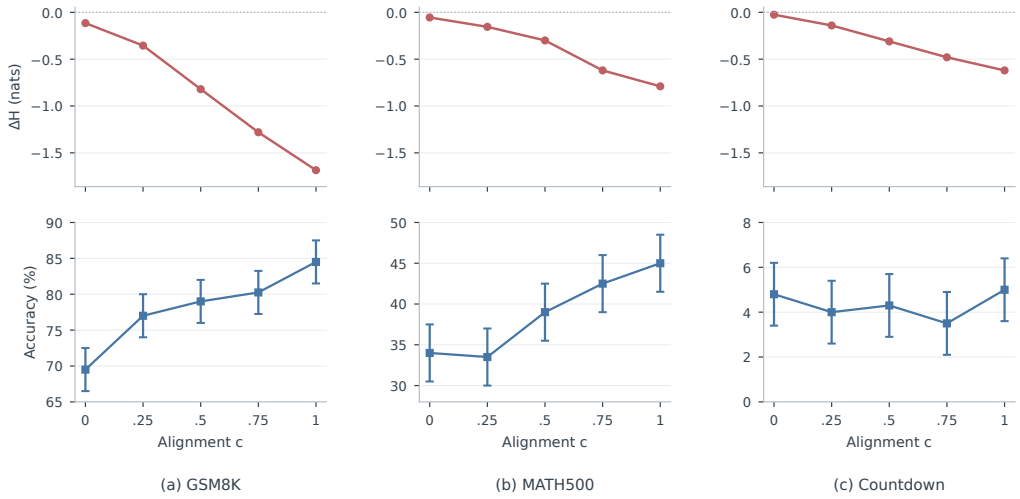


Figure 3: **Alignment reduces entropy; accuracy trends depend on the task.** Tülu-3/Llama on GSM8K, MATH-500, and Countdown. Top: teacher-forced ΔH. Bottom: accuracy at a 4096-token cap, with dataset-specific scales. Each $c < 1$ uses two direction draws; $c = 1$ is deterministic and agrees with the GSM8K truncated comparator to within 0.0009 nats. Lines connect sampled alignments, not training checkpoints; neither endpoint is SFT.

Neither end of the sweep is the full SFT-to-RLVR path. At $c = 0$, the update still has the full prescribed norm, so the model is not SFT. At $c = 1$, the model receives the rescaled top-24 update, not the full RLVR update. On GSM8K, this final point gives $\Delta H = -1.68453$ nats, within 0.0009 nats of the truncated reference's -1.6854. The full update gives -1.1136 nats. Appendix E provides the numerical consistency check.

3.4 Entropy tracks alignment; accuracy depends on the task

Lower entropy without consistent accuracy gains. Entropy falls as alignment increases on all three datasets (Figure 3). Accuracy does not follow one common pattern. It rises on GSM8K and rises overall on MATH-500 after a small initial dip, but remains low and approximately flat on Countdown. Pointing more closely toward the learned direction therefore lowers entropy consistently, but does not improve accuracy on every task.

This association does not show that lower entropy causes higher accuracy: both can respond to the same weight change, and we have only two draws at each $c < 1$. The 4096-token limit can also affect accuracy if an intervention changes response length. These scores are therefore separate from the 8192-token evaluation in Appendix A. We next ask the converse question: can accuracy improve without the entropy decrease?

4 Finding Accuracy Gains Without the Observed Entropy Reduction

The learned direction lowers entropy more than the matched random controls. But must an accurate model follow that direction? As shown in the graph in Appendix D, a simple linear interpolation from SFT to RLVR checkpoints shows that even a small movement away from SFT to RLVR weights increases performance a lot and gives an accuracy close to that of RLVR checkpoint itself. This shows that the direction learned by RLVR is essential for accuracy as well as entropy collapse, its magnitude less so. The graph suggests that around the RLVR checkpoint, there are weight profiles that have the same accuracy. To test this, we use the RLVR update to select matrices for a search, then discard its direction. Starting from SFT, we try random changes only to those matrices and select a

Table 1: **Selecting the most entropy-reducing perturbations does not make lower entropy correspond monotonically to higher test accuracy.** This auxiliary Tülu-3/Llama-3.1 8B search ranks 250 localized perturbations by entropy on 500 GSM8K training prompts, then evaluates accuracy on 500 untouched test prompts. Entropy is truncated top-20 Shannon entropy along each model's own greedy rollout. The majority rows report the mean training-set ΔH of the selected models; Appendix G gives the full protocol.

Lineage+Dataset	checkpoint/selection	test accuracy	H_{train}
Tülu-3/GSM8K	SFT	61.60%	0.4056 nats ($\Delta H = +0.0000$)
	top-5 weight ensemble	62.40%	0.3770 nats ($\Delta H = -0.0286$)
	RLVR	76.60%	0.2516 nats ($\Delta H = -0.1540$)
	$K = 2$ majority	58.20%	mean selected $\Delta H = -0.0198$
	$K = 5$ majority	69.20%	mean selected $\Delta H = -0.0189$
	$K = 12$ majority	73.20%	mean selected $\Delta H = -0.0149$
	$K = 25$ majority	73.00%	mean selected $\Delta H = -0.0114$

candidate based on task performance. The search uses *where* RLVR changes the weights, but not *how* it changes them.

We use Gan & Isola's 2026b approach to search for task improvements using random weight perturbations, but we restrict it to RLVR-selected matrices and we do not optimize for entropy.Finding an accurate candidate with SFT-like entropy would show that an observed accuracy gain need not reproduce RLVR's entropy decrease.

To select matrices from the RLVR update, for each projection matrix i, we compute $\Delta W_i = W_i^{\mathrm{RL}} - W_i^{\mathrm{SFT}}$ and use the update magnitudes to identify a set of 30 strongly changed matrices, $\mathcal{I}_{30}$. Figure 4(a) shows the selection, colored by relative update norm.

Searching locally around SFT. We add independent, isotropic Gaussian noise with standard deviation σ to the entries of the matrices in $\mathcal{I}_{30}$ and leave all other weights at SFT:

$$W_i' = W_i^{\mathrm{SFT}} + \begin{cases} \epsilon_i, & (\epsilon_i)_{ab} \sim \mathcal{N}(0, \sigma^2), \quad i \in \mathcal{I}_{30}, \\ 0, & i \notin \mathcal{I}_{30}. \end{cases} \tag{4}$$

Among the model weights found in this search, we call the candidate with the highest selection-split accuracy *Top-$\mathcal{I}_{30}$*. Unlike the earlier controls, this search fixes neither alignment nor update norm relative to RLVR. The noise scale σ sets the typical distance from SFT. If D_{30} is the number of perturbed weight entries, then $\mathbb{E}\|\epsilon\|_F^2 = D_{30}\sigma^2$. This is a search over all entries of the selected matrices, not another norm-matched direction test in the singular support.

Held-out evaluation. Candidate ranking uses only the search split. After selecting Top-$\mathcal{I}_{30}$, we freeze its weights and evaluate Top-$\mathcal{I}_{30}$, SFT, and RLVR on a separate set of untouched evaluation problems. The 61%, 76%, and 75% accuracy estimates reported below therefore come from problems that were not used to choose the perturbation. This differs from the broader Appendix A search, which reuses its selection problems for evaluation.

Close accuracy point estimates with different entropy. On the untouched evaluation problems, Top-$\mathcal{I}_{30}$ has a 76% accuracy point estimate, compared with 61% for SFT and 75% for RLVR (Figure 4b). Its mean entropy is 0.4120 nats, close to SFT's 0.4102, whereas RLVR's is 0.2350 (panel c). Thus, Top-$\mathcal{I}_{30}$ gains 15 percentage points of observed accuracy while barely changing entropy ($\Delta H = +0.0018$ nats). RLVR gains 14 points while lowering entropy substantially ($\Delta H = -0.1752$ nats). The 76% and 75% estimates are close, but they do not establish equivalence or that random search outperforms RLVR. They show that *an accuracy gain need not reproduce RLVR's observed entropy reduction*. We have not run repeated searches or a matched search over other matrix sets, so we cannot say that these locations are uniquely useful or optimal. Appendix A reports a separate, broader perturbation search and compares entropy distributions.

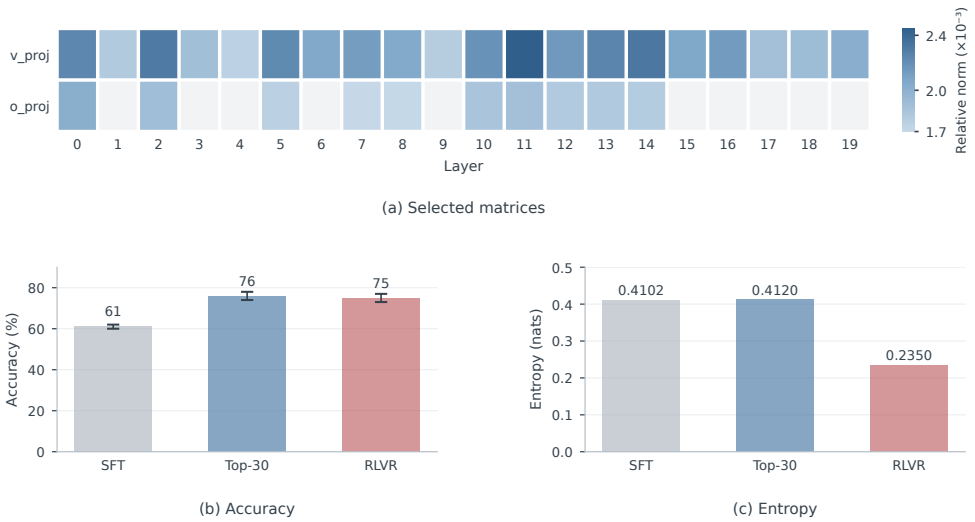


Figure 4: **A localized perturbation retains SFT-like entropy at an accuracy point estimate close to RLVR's.** (a) RLVR's update magnitudes identify 30 attention value and output projection matrices; Gaussian perturbations are applied to all entries of these matrices, and all other parameters remain at SFT. (b) On untouched evaluation problems, accuracy point estimates are 61% for SFT, 76% for Top-$\mathcal{I}_{30}$, and 75% for RLVR. (c) Mean entropies are 0.4102, 0.4120, and 0.2350 nats, respectively. The 76% versus 75% comparison is descriptive, not an equivalence or superiority claim, and this matrix-localized search is separate from the SVD-control support.

5 How Does RLVR Lower Entropy?

We now examine the output scores to understand how the full RLVR model produces that decrease. A simple explanation would be that RLVR makes SFT's predictions more confident without changing its token preferences. Multiplying all logits at a position by a positive number $a > 1$ does exactly this: it preserves token order but makes the softmax distribution more concentrated. This is *temperature scaling*, with temperature $T = 1/a$. We test whether such a rescaling explains the observed entropy decrease, or whether token-specific changes are needed.

5.1 Is RLVR simply sharpening SFT's predictions?

At each position, let z_S and z_R be the SFT and full-RLVR logit vectors, with one score per vocabulary token.[2] We fit a scale a and a shared offset b by least squares and define the remaining difference as r:

$$z_R = az_S + b\mathbf{1} + r, \tag{5}$$

We first evaluate just $z_{\text{temp}} = az_S + b\mathbf{1}$. These modified logits form a *counterfactual*: the predictions we would obtain if RLVR made only the fitted rescaling. If simple sharpening explains the entropy reduction, this counterfactual should reproduce it.

The fitted affine component increases entropy. On OLMo-3, the median scale is $a = 0.94$, and only 8% of positions have $a > 1$. The fit is close, with $R^2 = 0.94$; this coefficient of determination measures how much of the variation across RLVR token scores the fit explains. Yet the rescaling raises mean entropy from 1.294 to 1.642 nats ($+0.35$), while the actual RLVR model lowers it to 1.193 nats (about -0.10). Tülu-3/Llama shows the same reversal: the fitted rescaling changes entropy by $+0.85$ nats, while RLVR changes it by -1.29. The norm-matched within-support and orthogonal controls give $a = 1.000$ to the reported precision. The fitted component therefore flattens predictions rather than sharpening them.

[2]This experiment uses an earlier set of 100 fixed sequences and the full checkpoints, not the rescaled truncated updates. Its absolute entropy values must therefore be kept separate from Table 4.

5.2 TOKEN-SPECIFIC CHANGES EXPLAIN THE ENTROPY DECREASE

Since the fitted rescaling raises entropy, the decrease requires the residual r in Equation 5. We call its effect *re-ranking*: changing relative token scores beyond a shared rescaling. Which tokens account for this reranking effect? At each position, we order tokens by SFT probability. Starting from z_{temp}, we add back the residual only for the top-k tokens, or for nonoverlapping groups of ranks, and recompute entropy. Here k counts vocabulary entries, not singular components. On both lineages, changes among roughly the ten highest-probability SFT vocabulary entries recover 90% or more of the measured entropy reduction. We call these high-probability entries the *head* of the distribution; the many lower-probability entries form its *tail*. Changes in the tail can partly cancel one another but have much less effect on entropy.

This does not mean that the residual changes only a few token scores. On OLMo, its *participation ratio*—a measure of how many coordinates contribute appreciably to a vector—is approximately 6.2×10^4 tokens per position. The 100 tokens with the largest positive residuals account for less than 1% of the total positive residual magnitude. Most score-change magnitude is therefore spread across low-probability tokens, even though most of the entropy effect comes from high-probability tokens (For details see Section H and Figure 8 there).

6 DISCUSSION AND CONCLUSION

Our controlled interventions show that update orientation matters for entropy. The truncated RLVR update lowers entropy much more than random alternatives with the same singular support, singular values, and norm. This isolates orientation from update size and available matrix components: norm-matched noise alone would not provide the same control. W

Accuracy gains, however, need not reproduce the observed entropy reduction. In a separate intervention family, Top-$\mathcal{I}_{30}$ perturbs all entries of matrices selected by RLVR rather than remaining in the singular support. It attains a held-out accuracy point estimate close to RLVR's while retaining SFT-like entropy. This is one counterexample on an untouched evaluation split, not evidence of statistical equivalence or superiority. Together, the experiments distinguish *where useful weight changes can be found* from *the particular orientation that reduces entropy*.

The logit analysis explains how the full RLVR model produces this reduction. A fitted affine transformation accounts for most variation in the logits but increases entropy, so uniform sharpening is not the mechanism. Token-specific residual changes are widespread, yet changes among high-probability vocabulary entries recover most of the entropy effect. Supplementary analyses further show that attention and MLP updates both contribute and reinforce one another, while a late-layer component provides a local example of interpretable token-score changes (Appendices J and I). Overall, RLVR's entropy reduction is a directional and token-specific effect, whereas its observed accuracy gain can also be reached without reproducing that reduction.

We have reproduced these behaviors on two model lineages. However the evidence we have adduced does not establish *why* RLVR learns this orientation or whether orientation sensitivity is unusually specific to RLVR entropy rather than a generic property of trained functional directions. While our research points to an entropy-specific mechanism for RLVR in weight space, further evidence is needed to establish this definitively.

7 LIMITATIONS

We analyze final checkpoints from two lineages, not repeated RLVR training runs. We therefore do not know whether entropy-lowering orientations recur across training seeds, reward datasets, or recipes; the key next experiment is to repeat RLVR from the same SFT checkpoint and compare update alignments. Top-$\mathcal{I}_{30}$ uses one search run and one held-out split, so its close accuracy estimate does not establish equivalence or broad robustness. The Appendix A search reuses selection problems and does not test generalization.

Reproducibility statement

Sections 3–5 specify the controls, sample counts, Top-$\mathcal{I}_{30}$ search, and logit tests. The supplement records checkpoint and parameter-set definitions, seeds, norm and overlap checks, replay settings, and evaluation budgets in Appendices C, F, and K.

References

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, et al. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, and Matthias Plappert. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

DeepSeek-AI et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Yulu Gan and Phillip Isola. RandOpt data release. GitHub repository, `https://github.com/sunrainyg/RandOpt`, 2026a.

Yulu Gan and Phillip Isola. Neural thickets: Diverse task experts are dense around pretrained weights. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026b.

Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. Rethinking entropy interventions in rlvr: An entropy change perspective. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 31105–31133, 2026.

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*, volume 34, 2021.

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*, 2023.

Jaeeun Jang, Hansle Lee, and Sangmin Kim. A few bad apples spoil the bunch: Preventing global entropy collapse driven by a small set of tokens in llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13134–13154, 2026.

Renren Jin, Pengzhi Gao, Yuqi Ren, Zhuowen Han, Tongxuan Zhang, Wuwei Huang, Wei Liu, Jian Luan, and Deyi Xiong. Revisiting entropy in reinforcement learning for large reasoning models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25300–25322, 2026.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, et al. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Llama Team, AI at Meta. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ekdeep Singh Lubana, Eric J. Bigelow, Robert P. Dick, Hidenori Tanaka, and Tomer D. Ullman. Mechanistic mode connectivity. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Zeyue Qiu et al. ES-at-Scale: Evolution strategies at scale. GitHub repository, `https://github.com/VsonicV/es-at-scale`, 2026.

Team OLMo et al. OLMo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, et al. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Hao Ye, Jisheng Dang, Junfeng Fang, Bimei Wang, Yizhou Zhang, Ning Lv, et al. On the implicit reward overfitting and the low-rank dynamics in rlvr. *arXiv preprint arXiv:2605.06523*, 2026.

Chencheng Zhu. Orientation, not magnitude: The causal structure of task-vector interference in merged language models. *arXiv preprint arXiv:2608.11797*, 2026.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, et al. The path not taken: Rlvr provably learns off the principals. *arXiv preprint arXiv:2511.08567*, 2025.

Hanqing Zhu, Wenyan Cong, Zhizhou Sha, Sagnik Mukherjee, Xinyuan Song, David González-Martínez, et al. Iso: An rlvr-native optimization stack. *arXiv preprint arXiv:2607.19331*, 2026.

A Does Accuracy Improvement Imply the Observed Entropy Reduction?

We test the accuracy–entropy relationship with a second search that perturbs a broader set of SFT weights than Top-$\mathcal{I}_{30}$. We also compare the distribution of entropy across generated responses. This experiment is separate from the 30-matrix search in Section 4.

Following Neural Thickets (Gan & Isola, 2026b), we draw 300 Gaussian perturbations around SFT over the weight set used in our original diagnostic. We select the candidate with the highest GSM8K *reward*, defined here as the fraction of answers scored correct. We then reproduce its weights exactly from its stored random seed ($\sigma = 0.001$) and compare it with SFT and RLVR on the same 200 GSM8K problems. All three use the same scorer and an 8192-token generation limit. This common limit matters: the earlier 4096-token evaluation cut off more RLVR responses because they were much longer. These search settings and candidate counts do not describe Top-$\mathcal{I}_{30}$.

With the common limit, SFT, the selected perturbation, and RLVR obtain rewards of 0.890, 0.910, and 0.915. The perturbation and RLVR differ by just one correct answer out of 200, but their entropy changes have opposite signs: $+0.024$ versus -0.101 nats. The perturbation also keeps response length close to SFT's, with mean/median lengths of $1128/668$ versus $1043/673$ tokens. RLVR responses are longer, at $2330/1822$ tokens. The selected candidate therefore improves measured reward without reproducing RLVR's entropy decrease or longer responses.

Selection and evaluation use the same 200 GSM8K problems, so this comparison does not establish generalization. Selecting the best of many candidates can overestimate performance on new problems. The small observed gap also does not establish statistical equivalence; no equivalence test was run.

Figure 5 compares mean next-token entropy across generated responses. On both lineages, RLVR shifts this distribution toward lower entropy, whereas the selected perturbation stays closer to SFT or shifts higher. These measurements use generated text, so they are not identical to the shared fixed-text entropy test. Together, they show that similar measured accuracy can coexist with different entropy and response lengths.

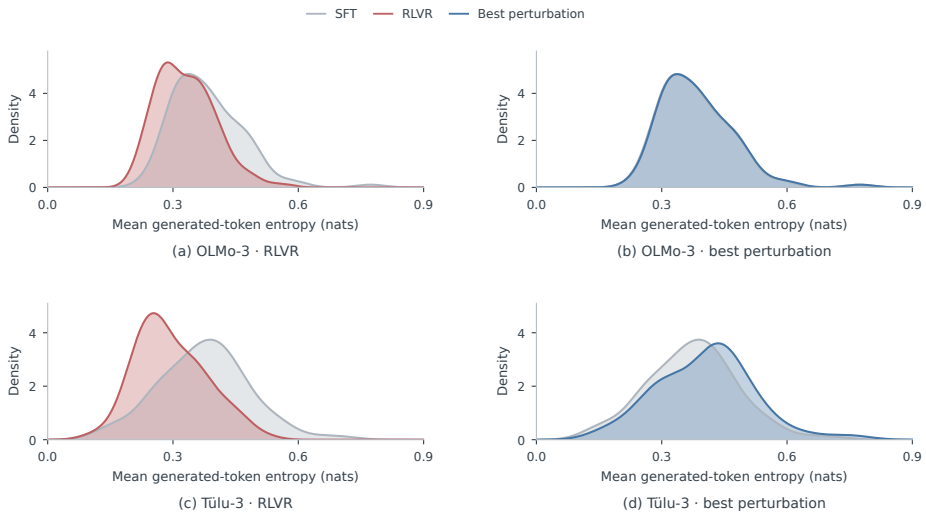


Figure 5: **Accuracy-improving perturbations do not reproduce RLVR's entropy shift.** Distributions of mean generated-token entropy for SFT (blue) versus RLVR (left column) and versus the selected best perturbation (right column), on OLMo-3 and Tülu-3/Llama. RLVR shifts both lineages toward lower entropy, whereas the perturbation remains close to SFT or shifts slightly higher. This broader perturbation comparison is distinct from the Top-$\mathcal{I}_{30}$ localized search in Section 4.

B An Initial Test With Random Weight Changes

Before designing the direction controls, we tested how small random weight changes affect each OLMo-3 checkpoint. We sampled 300 isotropic Gaussian perturbations per checkpoint, using noise scales $\sigma \in \{5\times10^{-4}, 10^{-3}, 2\times10^{-3}\}$. Reward was the fraction correct on 200 fixed GSM8K training problems with a 4096-token generation limit. Writing reward as J, we use ΔJ for the perturbed model's reward minus the original checkpoint's reward. This diagnostic measures performance on the training problems, not generalization.

Table 2: **Small random changes affect SFT and RLVR differently.** Results use the original 4096-token limit. "Strict" is the fraction of 300 perturbations that improve reward, $P[\Delta J > 0]$; "tie/beat" also counts ties, $P[\Delta J \geq 0]$. The reported z statistic standardizes the mean reward change relative to zero. RLVR's longer responses are more often cut off, so this table is not a fair comparison with the best perturbation.

Checkpoint	Reward	Mean ΔJ	z	Strict	Tie/beat
Base	0.815	-0.0299	-15.3	9.7%	13.7%
SFT	0.885	$+0.0085$	$+9.8$	64.3%	77.7%
RLVR	0.880	-0.0090	-7.5	26.7%	39.0%

Under this evaluation, SFT and RLVR have nearly equal reward but respond differently to perturbations: most improve SFT, while most reduce RLVR's reward. This suggested that accuracy alone misses differences in how the models behave under small weight changes. It did not identify which part of the RLVR update caused those differences. We therefore use it as motivation, rather than as a main result; related random-perturbation analyses are developed by Gan & Isola (2026b).

The best perturbation appears better than RLVR at 4096 tokens (0.930 versus 0.880 reward), but that advantage disappears when longer RLVR responses are allowed to finish. Reproducing the same candidate and evaluating all models with an 8192-token limit gives rewards of 0.890 for SFT, 0.910 for the perturbation, and 0.915 for RLVR. The last two differ by one correct answer out of 200, not enough to establish statistical equivalence. The perturbation still lacks RLVR's entropy decrease and

longer responses (Figure 5). As noted above, reusing the selection problems for evaluation prevents a generalization claim.

C BASIC SFT-RLVR CHECKPOINTS

Table 3 below shows that RLVR makes a much smaller update than SFT. The aggregate relative norm is approximately $156\times$ smaller on OLMo-3 and $22\times$ smaller on Tülu-3/Llama. RLVR also has lower effective rank: approximately 19 versus 54 on OLMo-3, and 17 versus 45 on Tülu-3/Llama. Changes in the weights' singular values and leading subspaces are small as well.

Table 3: **RLVR updates are smaller and more concentrated than SFT updates.** SFT denotes base-to-SFT and RLVR denotes SFT-to-RLVR. Values are aggregates over 224 attention and MLP projection matrices; effective rank measures spectral concentration, not exact algebraic rank.

	OLMo-3		Tülu-3/Llama	
Metric	SFT	RLVR	SFT	RLVR
Relative update norm	0.250	0.0016	0.0269	0.00123
Spectral drift	0.039	2.1×10^{-5}	4.6×10^{-4}	1.2×10^{-5}
Principal rotation	0.101	7.0×10^{-6}	1.7×10^{-3}	3.6×10^{-6}
Effective rank	≈ 54	≈ 19	≈ 45	≈ 17

D TESTING STRAIGHT-LINE PATHS BETWEEN CHECKPOINTS

We evaluate weighted averages of checkpoint weights, $\theta(\alpha) = (1-\alpha)\theta_A + \alpha\theta_B$, from base to SFT and from base to RLVR. Here $\alpha = 0$ is the base checkpoint, $\alpha = 1$ is the trained checkpoint, and values between them trace a straight line in weight space. The tested interior points do not show a sharp drop in reward (Figure 6). Beyond $\alpha = 1$, the apparent drop on the base-to-RLVR path at 4096 tokens disappears when the same points are evaluated at 8192 tokens. It therefore reflects cut-off responses rather than a sudden loss of task ability along that path. The low score at $\alpha = 0$ is also affected by evaluating the base model with a reasoning chat template it was not trained to use.

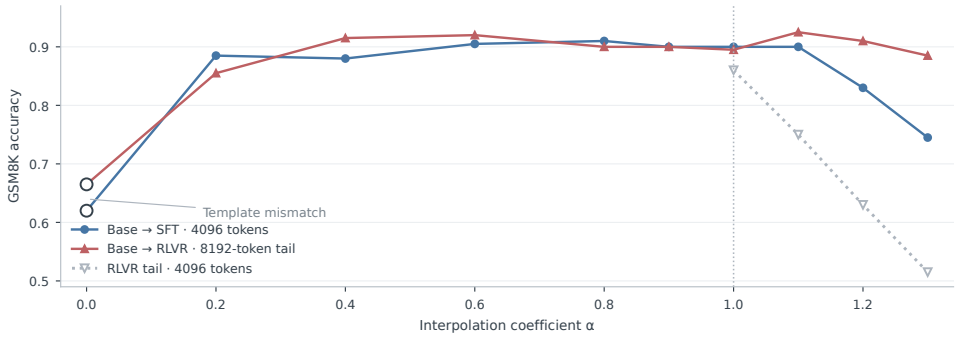


Figure 6: **Reward remains high over broad portions of the tested straight-line paths.** Paths run from base to SFT and from base to RLVR. Annotations mark the chat-template mismatch at $\alpha = 0$ and cut-off responses beyond the trained checkpoint under the 4096-token limit. The absence of a sharp drop on these paths does not establish how the checkpoints are connected elsewhere in weight space.

E CONTROLLED RANK SWEEP AND ENDPOINT CONSISTENCY

What is held fixed. Table 4 compares truncation ranks $k \in \{8, 24, 48\}$ and update families on the same fixed GSM8K text for each lineage. At each rank, the truncated learned update and spectrum-

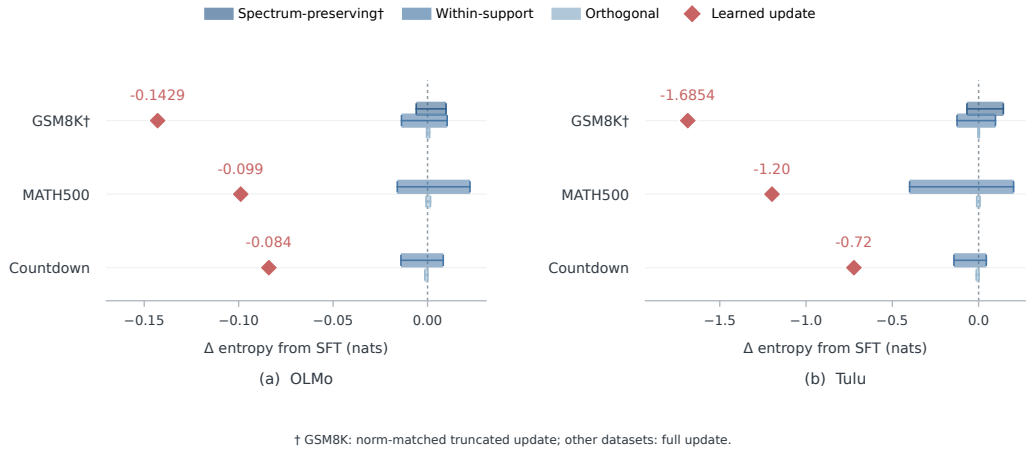


Figure 7: **Only the GSM8K row isolates orientation at fixed support, spectrum, and norm.** For both lineages, the GSM8K (†) comparison uses the norm-matched truncated update at $k = 24$ and includes the spectrum-preserving control. The MATH-500 and Countdown rows use the earlier full-update comparator with within-support and orthogonal controls, so they provide broader context but do not isolate orientation under the same matched design.

Table 4: **The learned orientation reduces fixed-text entropy at every tested rank.** Teacher-forced ΔH from SFT (nats) on 100 fixed GSM8K sequences. Every matrix update is rescaled to $\|\Delta\|_F$. Parentheses give standard deviations across 20 random interventions at $k = 24$ and 10 at $k = 8, 48$. The truncated and full updates are deterministic point estimates.

OLMo-3: $H(\text{SFT}) = 1.2939$; full-update $\Delta H = -0.1022$ (context)				
k	Truncated	Spectrum-preserving	Within support	Orthogonal
8	-0.1382	-0.0216 (0.0229)	$+0.0070$ (0.0227)	-0.0002 (0.0009)
24	-0.1429	$+0.0019$ (0.0079)	-0.0017 (0.0121)	$+0.0003$ (0.0008)
48	-0.1420	$+0.0003$ (0.0054)	$+0.0007$ (0.0060)	$+0.0002$ (0.0010)
Tülu-3/Llama: $H(\text{SFT}) = 2.8680$; full-update $\Delta H = -1.1136$ (context)				
k	Truncated	Spectrum-preserving	Within support	Orthogonal
8	-1.5669	-0.1040 (0.1584)	$+0.1312$ (0.1370)	-0.0009 (0.0044)
24	-1.6854	$+0.0378$ (0.1055)	-0.0138 (0.1114)	$+0.0004$ (0.0040)
48	-1.6459	$+0.0001$ (0.0613)	-0.0087 (0.0851)	$+0.0003$ (0.0049)

preserving rotations share support, norm, and singular values. The SFT baselines are 1.2939 nats for OLMo-3 and 2.8680 for Tülu-3/Llama. The Tülu-3/Llama alignment sweep uses the same text and gives $H(\text{SFT}) = 2.86803$, which agrees at the table's precision. The earlier logit and pathway experiments use a separate evaluation; their entropy baselines must not be replaced with these values.

Rank sensitivity. The learned update produces a similar entropy decrease across the tested ranks, but the variation across random controls changes. For spectrum-preserving controls, the standard deviation falls from 0.0229 at $k = 8$ to 0.0054 at $k = 48$ on OLMo-3, and from 0.1584 to 0.0613 on Tülu-3/Llama. At $k = 8$, their mean changes are slightly negative (-0.0216 and -0.1040), so random directions are not always without effect. Each mean is within one draw-level standard deviation of zero, and the learned update lowers entropy much more. The result is a consistent ordering at the tested ranks, not a claim about every possible k.

Rescaling and alignment-endpoint agreement. Let $\widehat{d} = \Delta_k / \|\Delta_k\|_F$. At $c = 1$, the alignment construction gives $\|\Delta\|_F \widehat{d} = \mathcal{N}_\Delta(\Delta_k)$, exactly the truncated reference update. The Tülu-3/Llama GSM8K results confirm this on the same text: the SFT baselines agree (2.86803 versus 2.8680 nats),

Table 5: **The learned updates reduce free-running entropy across three tasks and two lineages.** Mean next-token entropy along each model's own greedy trajectory (100 prompts, 256 generated tokens, $k = 24$). $H(\mathrm{SFT})$ is absolute entropy; all other entries are ΔH from SFT in nats. Parentheses give standard deviations across four random interventions. Full and truncated updates are deterministic point estimates.

Lineage	Dataset	$H(\mathrm{SFT})$	Full RLVR	Truncated	Spectrum-pres.	Within	Orthogonal
OLMo-3	GSM8K	0.3429	−0.0532	−0.0742	−0.0032 (0.0046)	+0.0075 (0.0049)	+0.0015 (0.0030)
	MATH-500	0.2991	−0.0477	−0.0756	+0.0028 (0.0041)	+0.0050 (0.0041)	+0.0043 (0.0020)
	Countdown	0.5656	−0.1620	−0.2486	−0.0043 (0.0080)	−0.0091 (0.0035)	−0.0035 (0.0020)
Tülu-3	GSM8K	0.4102	−0.1752	−0.2049	−0.0065 (0.0097)	+0.0111 (0.0123)	−0.0016 (0.0015)
	MATH-500	0.3523	−0.1339	−0.1617	−0.0033 (0.0175)	+0.0084 (0.0091)	+0.0030 (0.0041)
	Countdown	0.5229	−0.3249	−0.1192	−0.0251 (0.0470)	−0.0093 (0.0266)	+0.0037 (0.0021)

and $\Delta H(c = 1) = -1.68453$ differs from the rank-24 reference's -1.6854 by less than 0.0009 nats (about 0.05%). This small difference is consistent with numerical variation from bfloat16, the reduced-precision floating-point format used in the computation. The check resolves an earlier discrepancy between these experiments. It does not make the rescaled truncated update identical to full RLVR.

Relating weight changes to logit changes. The matched controls test update direction while holding size and structure fixed. The residual analysis instead tests the full model's output scores on the earlier evaluation text. We have not applied that analysis to every rescaled truncated model. Doing so would test whether truncation and rescaling preserve the same pattern of token-score changes.

Free-running replication. Figure 2 plots Table 5. The learned updates also lower entropy more than random controls when each model generates its own text. On Tülu-3/Countdown, the full update has a larger effect than the truncated update, reversing their order in the other cases. The size of the effect can therefore depend on both the weights and the text they generate. This short-generation test does not measure diversity across unrestricted responses or remove length-limit effects from the separate 4096-token accuracy sweep.

F Which Weights Each Analysis Includes

The analyses use different sets of weights. A *tensor* is a weight array; matrices are two-dimensional tensors. We distinguish two sets:

- **224 matrices:** the attention and MLP projections used for SVD measurements and direction controls. Across these matrices, the raw OLMo update norms are 467.5 for base-to-SFT and 2.81 for SFT-to-RLVR, a $166\times$ difference.
- **355 weight tensors:** a broader set that also includes embeddings and other weights. We use it to measure how update *energy*, the squared norm, is divided across model components. MLP projections account for 64% of RLVR update energy and attention for 31%; most of the remainder is in embeddings. Relative movement per parameter is similar for MLP and attention (0.0015 versus 0.0017).

The counts differ because the analyses ask different questions. The SVD experiment studies projection matrices; the component-level accounting includes additional weights.

The original perturbation experiment uses the runtime's *fused layout*, in which some parameter arrays are stored together. Over that weight set, it reports $\|\Delta_{\mathrm{RL}}\|_F = 2.74$ and $D \approx 6.48 \times 10^9$ weight entries. The root-mean-square (RMS) update, $\|\Delta_{\mathrm{RL}}\|_F/\sqrt{D}$, is 3.40×10^{-5}. This norm must be kept separate from the 2.81 measured over the 224 SVD matrices. The main control experiment avoids this mismatch by matching update norms separately for each of those 224 matrices.

Tülu-3/Llama shows the same ordering as OLMo-3 for relative norm, spectral drift, subspace rotation, and effective rank (Table 3). The size of the SFT–RLVR gap differs: approximately $156\times$ in relative norm on OLMo and $22\times$ on Tülu.

G Entropy-Selected Localized Perturbation Search

This section documents the auxiliary search summarized in Table 1. It uses the same type of matrix-localized perturbation as Section 4, but ranks candidates by entropy rather than accuracy. The experiment tests whether perturbations with the strongest entropy reduction also give the largest accuracy gains; it is not a norm-matched intervention in the SVD singular support.

Models, data, and inference. The base checkpoint is `OpenFinAI/Tulu-3-8B-SFT`, built on Llama-3.1 8B, and the comparison checkpoint is `OpenFinAI/Tulu-3-8B-RLVR`. We use 500 GSM8K training prompts for selection and a disjoint set of 500 test prompts for accuracy evaluation. The experimental code uses plain concatenated prompts and checks extracted numeric answers with the GSM8K value-level checker. Inference uses six vLLM V1 engines with tensor parallelism one, bfloat16 weights and computation, and global seed 42. The reported run used the CUDA 12.9 container `vllm-cu129.sif`.

Selecting the perturbation mask. For every matched SFT and RLVR parameter matrix W_i, we compute the relative RLVR update magnitude

$$r_i = \frac{\|W_i^{\mathrm{RLVR}} - W_i^{\mathrm{SFT}}\|_F}{\|W_i^{\mathrm{SFT}}\|_F}. \tag{6}$$

We rank matrices by r_i and perturb only the top 30. The selected set is dominated by attention value and output projections (`v_proj` and `o_proj`); every parameter outside this mask remains at its SFT value. This selection identifies where RLVR moved most in relative norm but does not preserve the learned update direction.

Perturbation population. The population contains $P = 250$ perturbations. Using `numpy.random.default_rng(42)`, each candidate receives a distinct seed from $[0, 2^{31})$ and a noise scale sampled from $\{0.0005, 0.001, 0.002\}$. For each masked matrix,

$$W_i' = W_i^{\mathrm{SFT}} + \sigma\epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, I), \tag{7}$$

where the bfloat16 noise is generated on the parameter's device with a `torch.Generator` initialized from the candidate seed. Applying the same seed and scale replays a candidate exactly in the reported software and hardware configuration. After each rollout, the inverse update restores the SFT weights before the next candidate is evaluated.

Free-running entropy metric. Every model generates its own greedy trajectory with temperature zero, a 1024-token limit, and decoding seed 42. At each generated token, vLLM returns the top 20 log-probabilities $\ell_1, \ldots, \ell_{20}$, the maximum supported by this runtime. After clipping them to $[-25, 0]$, we compute the truncated Shannon entropy

$$\widehat{H}_t = -\sum_{j=1}^{20} \exp(\ell_j)\ell_j. \tag{8}$$

We average first over generated tokens and then over the 500 training prompts. The computation also records the returned probability mass, $\sum_j \exp(\ell_j)$, as a truncation check; values near one indicate that top-20 entropy closely approximates full-vocabulary entropy for these peaked predictions. This free-running quantity is distinct from the teacher-forced entropy used in the fixed-text direction controls.

Selection and evaluation. We evaluate SFT, all 250 perturbations, and RLVR on the training prompts, then rank perturbations by mean entropy in ascending order. The lowest-entropy candidates define sets of size $K \in \{2, 5, 12, 25\}$. Each of the top 25 models is evaluated on the untouched test prompts, and plurality voting combines their extracted answers at each K. We also form a top-5 weight ensemble by averaging the five selected perturbation updates,

$$W_i^{\mathrm{ens}} = W_i^{\mathrm{SFT}} + \frac{1}{5}\sum_{k=1}^{5} \sigma_k \epsilon_{i,k}, \tag{9}$$

and evaluate this single merged model on both splits. Thus, entropy determines selection on the training split, whereas all reported accuracies in Table 1 are measured on held-out test problems.

Table 6: **Exact settings for the entropy-selected perturbation search.** All candidates use the same matrix mask, generation procedure, and seeded population; only their measured training-rollout entropy determines the ranking.

Setting	Value
Models	Tülu-3 8B SFT / RLVR
Dataset	GSM8K; 500 train / 500 held-out test
Perturbed parameters	Top 30 matrices by relative RLVR-update norm
Population and scales	250; $\sigma \in \{0.0005, 0.001, 0.002\}$
Precision and seed	bfloat16; global seed 42
Decoding	Greedy ($T = 0$); maximum 1024 generated tokens
Selection score	Mean top-20 free-running entropy on train
Selected sets	$K \in \{2, 5, 12, 25\}$
Weight ensemble	Equal average of the top 5 perturbation updates
Evaluation	Individual, ensemble, and majority-vote accuracy on test

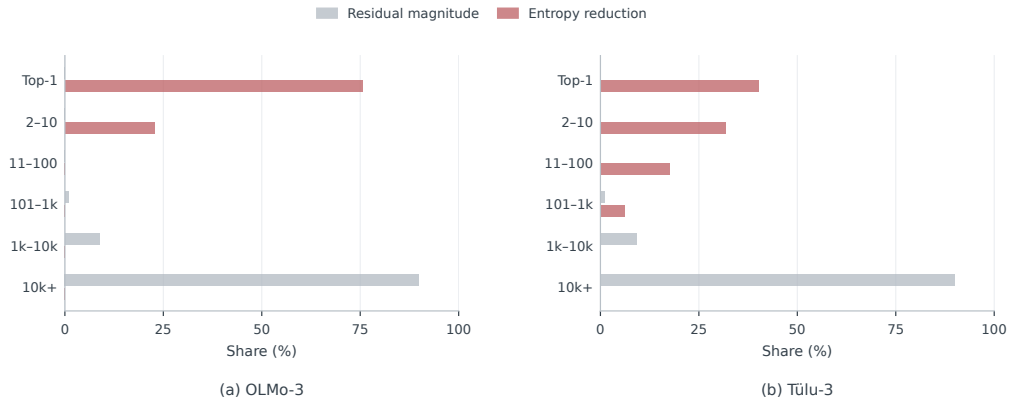


Figure 8: **Score changes are widespread, but high-probability tokens drive the entropy decrease.** Tokens are grouped by their SFT probability rank. Gray bars show each group's share of residual-logit magnitude; colored bars show its effect on entropy. In both lineages, low-probability tokens account for most residual magnitude, while high-probability tokens account for most entropy reduction.

Interpretation and reproducibility. The two lowest-entropy candidates underperform SFT after voting, while the $K = 12$ vote reaches 73.2% test accuracy with a mean selected entropy change of only -0.0149 nats, compared with RLVR's -0.1540 nats. Entropy ranking can therefore identify a useful collection of alternative predictions, but stronger entropy reduction is not monotonically associated with higher accuracy. Seeds control population construction, parameter noise, and decoding. Reproducing the reported values most closely additionally requires the same bfloat16 vLLM/CUDA stack because GPU kernels can otherwise introduce small numerical differences.

H Tables for Reranking

The explanation concerns relative scores, not a literal transfer of probability between fixed token pairs: softmax renormalizes all probabilities after the scores change. Changes to tokens that already have substantial probability have the largest entropy effect. This pattern holds across both lineages, but the token identities differ. OLMo and Tülu emphasize different token classes and even change `Wait` in opposite directions. The shared result is the importance of high-probability tokens, not a universal list of words that RLVR promotes or suppresses.

I Reading a Late-Layer Update in Token Scores

To explore which tokens a late-layer update might affect, we map its left singular vectors to vocabulary scores. We use the final block's `mlp.down_proj`, which maps MLP outputs back into the *residual stream*: the shared token representation passed between transformer blocks. Its output vectors can be read in that space, unlike the input-side directions of the query, key, value, gate, and up projections.

The readout applies the learned gain of the final root-mean-square normalization layer (RMSNorm), then the *unembedding matrix*, which maps the final representation to token logits. Such a readout is often called a *logit lens*. Singular-vector signs are arbitrary, so we choose their signs using how the update acts on mean input activations. This makes positive and negative projected shifts follow the actual update rather than an arbitrary SVD sign.

Two checks support this approximate readout. First, applying the lens to a token's own unembedding direction gives a median rank of zero for that token, meaning it is ranked first under zero-based indexing. Second, equally sized random directions in the residual stream produce smaller peak logit shifts than RLVR singular components: means of 0.12 versus 0.78. In singular component, or *mode*, 3, scores rise for "So/therefore" variants and fall for "Wait" variants (Figure 9); another mode shows a related pattern in the opposite direction. This is a finding about one late-layer update, not a global "Wait-to-So" mechanism. Simple logit lenses are approximate; trained tuned lenses can give more faithful readouts of intermediate representations (Belrose et al., 2023; Elhage et al., 2021).

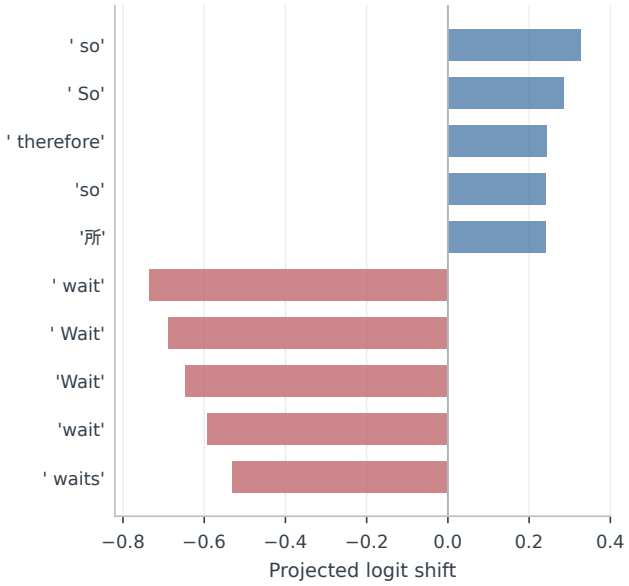


Figure 9: **One late-layer OLMo component favors conclusion variants over "Wait" variants.** After fixing the singular-vector sign using the update's action, the approximate readout raises scores for several conclusion-token variants and lowers scores for several "Wait" variants. This local example does not establish a global token-level mechanism.

J Separating Attention and MLP Updates

We split the OLMo update into attention projections (query, key, value, and output), MLP projections (gate, up, and down), and the remaining weights. We then apply the attention and MLP updates to SFT separately and together. Attention changes entropy by -0.050 nats and MLP by -0.049. Together they give -0.103, matching the full update (Figure 10). Both parts therefore make substantial contributions.

We also ask whether the two parts change predictions in similar or opposing ways. Kullback–Leibler (KL) divergence measures how much one probability distribution differs from another; for distributions p and q, $\mathrm{KL}(p\|q) = \sum_v p(v) \log[p(v)/q(v)]$. The joint update's KL divergence from SFT is 0.173, larger than the sum for the separate updates ($0.019 + 0.128 = 0.147$). Their effects on the distribution therefore reinforce rather than cancel.

A second measure, c_F, compares the attention and MLP logit changes using a correlation weighted by SFT's token probabilities. This is the *Fisher-metric correlation*: it measures alignment in the local probability geometry rather than giving every token score equal weight. Positive values indicate aligned changes. On OLMo, $c_F = +0.68$. Shuffling token positions breaks the correspondence between the two sets of changes and provides a reference distribution; the observed correlation is $z = 5.1$ standard deviations above its mean.



Figure 10: **Attention and MLP updates both lower entropy on OLMo.** Applying them together recovers the full update's entropy effect, and their changes in token probabilities reinforce each other. The relative cost in KL divergence per unit of squared weight change differs across lineages.

Tülu-3/Llama shows the same two findings. Attention and MLP change entropy by -0.831 and -0.595 nats, and together recover the full update's -1.294 change. Their effects reinforce more strongly ($C_{KL} = -0.99$, $c_F = +0.90$, $z = 5.8$). Here C_{KL} is the reported cancellation index comparing joint KL divergence with the sum of separate divergences; negative values indicate reinforcement rather than cancellation. However, the relative KL cost after normalizing by squared update norm, $\mathrm{KL}/\|\Delta\|_F^2$, reverses: attention has lower cost than MLP on OLMo and higher cost on Tülu. Both parts contribute and reinforce, but neither is universally more efficient.

A finer analysis applies individual singular components of selected OLMo matrices. Components that lower entropy and components that raise it appear throughout the singular-value ranking, not only among the largest values (Figure 11). For the sampled MLP matrices, the separate component effects roughly add up to the full matrix effect. Attention components interact more strongly, so their separate effects are less additive. This analysis covers only one lineage and a selected matrix set; it suggests a question for further study rather than a general result.

K Statistical and Implementation Details

Units and uncertainty. The fixed-text rank sweep uses 100 GSM8K sequences per lineage and $N = 20$ full-model draws per random family at $k = 24$, with $N = 10$ at $k = 8, 48$. Here N is the number of random interventions. The free-running check uses 100 prompts per dataset and four draws

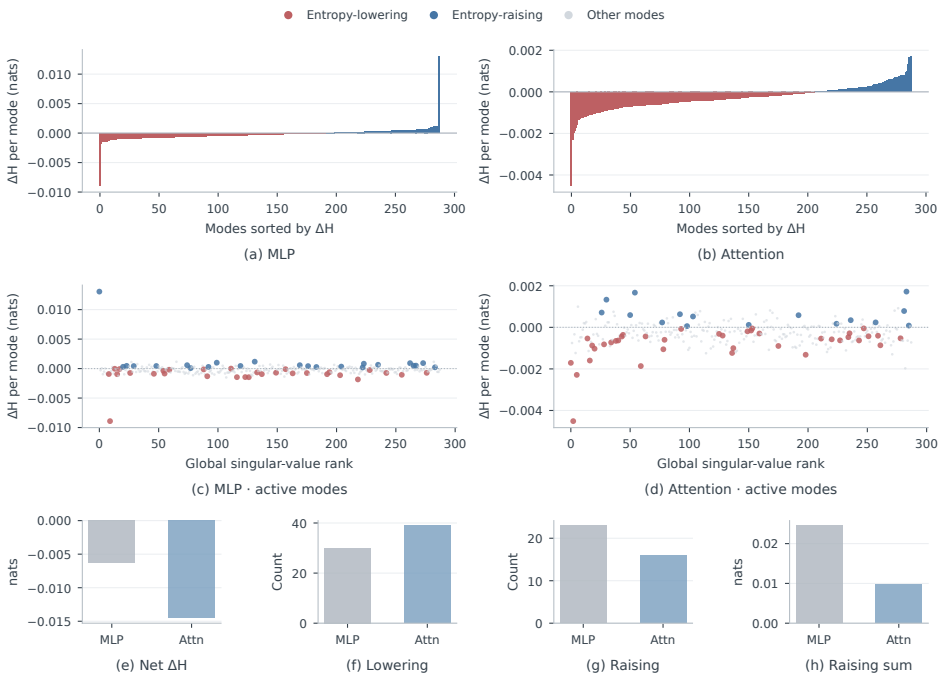


Figure 11: **Individual singular components can raise or lower entropy.** This OLMo analysis uses 24 matrices each from attention and MLP. Panels (a,b) order components by entropy effect; (c,d) show where components with an effect fall in the overall singular-value ranking. The remaining panels summarize net entropy change, counts of lowering and raising components, and summed raising effects. Components with entropy effects appear throughout the ranking in both groups.

per random family. Tables 4 and 5 report standard deviations across interventions, not confidence intervals or variation across tokens. The truncated and full updates have no random-intervention variation and are reported as point estimates. The alignment sweep uses the separate repetition counts below.

Norm and overlap checks. We generate controls separately for each matrix and rescale them to that matrix's RLVR update norm. Norms agree to four decimal places, and the orthogonal controls have overlap with the retained support on the order of 10^{-5}. For spectrum-preserving rotations, the unscaled coefficient singular values match $\mathrm{diag}(\Sigma_k)$ to four decimal places. The fraction of squared norm retained in $\mathcal{S}_k$ is 1.000000, and cosine alignment with the learned direction is on the order of 10^{-2}. Rescaling multiplies the singular values of both the reference and control by $\|\Delta\|_F/\|\Delta_k\|_F$, so their spectra still match afterward.

Alignment-sweep construction and replay. For Equation 3, the random seed at target c and repetition index j is $9000 + \lfloor 1000c \rfloor + j$. Before evaluation, the fraction of squared norm in $\mathcal{S}_k$ is one to six decimal places, and $\langle \widehat{e}, \widehat{d} \rangle_F \approx 10^{-17}$. The evaluated weights have cosine alignment within 10^{-5} of each target. The coefficient along the learned unit direction is c, so that component accounts for a fraction c^2 of the squared norm. We use two independent draws at each of $c = 0, 0.25, 0.5, 0.75$ and one deterministic update at $c = 1$. Entropy uses 100 fixed SFT-generated sequences; accuracy uses 200 held-out problems per dataset with a 4096-token limit. A deterministic update still has uncertainty in accuracy estimated from a finite problem set. Appendix E checks the final GSM8K point against the truncated reference.

Evaluation controls. Within each lineage, the fixed-text comparisons use the same prompts, chat template, sequences, positions, padding masks, and entropy code. No text is sampled during these

forward passes. The measurement is uncertainty in next-token predictions, not variation across generated responses.

Temperature and residual decomposition. At each position, we fit RLVR logits from SFT logits using a scalar slope and intercept. Subtracting the mean from each logit vector before fitting is equivalent, because a shared offset does not affect softmax. The temperature-only test uses the fitted scale and offset without the residual. The top-k and rank-group tests then add back residual entries selected by SFT probability rank. We compute participation ratios and residual-magnitude summaries separately at each position before aggregating.

Perturbation and generation budgets. The initial reward diagnostic uses 200 GSM8K training examples, 300 perturbations per checkpoint, and a 4096-token limit. The common-budget comparison reproduces the selected perturbation from its stored seed ($\sigma = 0.001$) and evaluates SFT, the perturbation, and RLVR with the same 8192-token limit and scorer. We also re-evaluate interpolation points with cut-off responses at 8192 tokens. Reproducing the random noise requires the same fused parameter layout used during the original run.