WHAT MERGING MISSES: THE DEFINING DIRECTION OF JOINT TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model merging rests on a working assumption: that reweighting a set of independently trained models can stand in for one model trained jointly on all their tasks. We question this assumption and show that it fails in a precise, structural way. Across 81 LLM settings, even the best reweighting of independent endpoints loses to joint training in 80 cases. This failure is not because the endpoints are aimed in the wrong directions. Their span already captures about 82% of the joint update, but it misses a single direction outside that span, and no reweighting of the endpoints can recover it, no matter how it is tuned. This missing direction follows a simple law: its energy, measured by the missing squared norm, scales with the number of task pairs rather than the number of tasks. As a result, the gap grows quadratically as tasks accumulate, and a law fit on a few task counts can predict the gap at counts held out from the fit. The missing direction also has a definite origin. Almost all of it comes from a path-dependent term that joint training builds as the trajectory is pulled from one task toward the next. When this term is read from the training path and added back, the lost performance returns; when its reverse, or a random direction of the same length, is added instead, it does not. What sets joint training apart, then, is not where independently trained runs end up, but what their weights never record: a contribution built from the interaction between tasks as they are learned together. Recombining separately trained results, however carefully they are weighted, cannot reconstruct it.

1 INTRODUCTION

Checkpoint-based model merging offers a compelling alternative to joint training: it combines independently trained models without revisiting their original data, but obtains this efficiency by reducing each training process to the final model it leaves behind (Wortsman et al., 2022; Ilharco et al., 2023). Much of the field's sophistication has consequently been devoted to increasingly powerful ways of manipulating these endpoints after training has finished (Yadav et al., 2023; Yang et al., 2024b). This endpoint-centered program inherits a premise it rarely states: that whatever value joint optimization creates by learning tasks together is already latent in the endpoints those tasks reach in isolation, waiting to be recombined. If this holds, merging is a problem of coefficients. If it fails, part of what joint training produces is not a property of where independent runs arrive at all, and no reweighting can recover it. We therefore ask not whether one merging rule beats another, but:

What does learning tasks together create beyond the outcomes of learning each task alone, and what would a merging interface need to preserve in order to recover it?

Our answer begins with a structural finding: what joint optimization creates is neither an artifact of poor coefficients nor an unstructured residue. Even when endpoint combinations are granted oracle-level coefficients, weights selected directly against the joint target, the best span solution still loses to joint training in 80 of 81 LLM configurations, and the shortfall grows rather than vanishes as more tasks are combined. The missing part is not distributed noise but a coherent, functionally consequential direction that lies outside the space the endpoints span; coefficient tuning therefore operates in a space from which this direction is systematically absent, and as task compositions grow it can become the dominant component of the joint update. What joint training produces is thus not reducible to where its constituent tasks individually arrive.

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 value of this observation is that the missing direction is lawful in its size and separable into what optimization creates and what a target must then supply. Its magnitude follows a single-parameter law born from pairwise coordination between tasks. Fitting this law on a few small compositions predicts the endpoint-external energy at larger, held-out task counts in both a controlled system and LLM fine-tuning, so the gap can be anticipated before those compositions are ever trained. Moreover, this direction is not an abstract residual. It can be read off the joint trajectory as a path-accumulated component that accounts for nearly all of the missing direction in parameter space, and reinjecting it into the endpoint solution restores the lost performance.

More broadly, when objectives interact, final parameters are not a sufficient description of learning: they record where optimization ends, but not all of what is created while it moves. For model merging, this shifts the object of study from choosing coefficients over fixed endpoints to preserving the coordination created during co-learning and calibrating its contribution to a target composition.

Our main contributions are summarized as follows:

- **Endpoint sufficiency:** We formulate endpoint sufficiency as an exact projection problem with an additive zero reference, exposing a structural limit of the canonical linear interface: joint updates systematically escape the isolated-task span, and even oracle coefficients stay functionally insufficient in 80 of 81 LLM configurations.
- **Predictive scaling:** We derive the single-parameter law $q_K/(1-q_K) = \gamma(K-1)$ from pairwise coordination accumulation and show that fitting it on smaller task combinations accurately predicts endpoint-external energy at held-out larger values of K across both controlled systems and LLM fine-tuning.
- **Path-based recovery:** We trace the endpoint-external component to a residual accumulated along the joint optimization path, which accounts for nearly all of the missing direction in parameter space. This shows the missing part is not an irreducible loss beyond the reach of the endpoints, but a concrete object that can be located and restored: reading this residual off the training trajectory and reinjecting it recovers the lost performance.

2 Independent Endpoints Are an Incomplete Basis for Joint Updates

Since merging only takes weighted combinations of the endpoint updates, whether it can recover what tasks learn together reduces to a single fixed subspace: the endpoint span. Any joint-update direction outside it is unreachable by every choice of coefficients.

Setup. We pair each joint run with its isolated endpoints, using a controlled five-task testbed for additive checks and an 81-cell LLM grid over three four-task suites spanning QA/reasoning, MMLU-Pro subjects, and P3 templates, with three backbones (Llama-3.1-8B (Grattafiori et al., 2024), Qwen2.5-1.5B, and Qwen2.5-7B (Yang et al., 2024a)). LoRA ranks, and seeds. Exact protocols, metrics, hyperparameters, and audits are deferred to Appendix A.

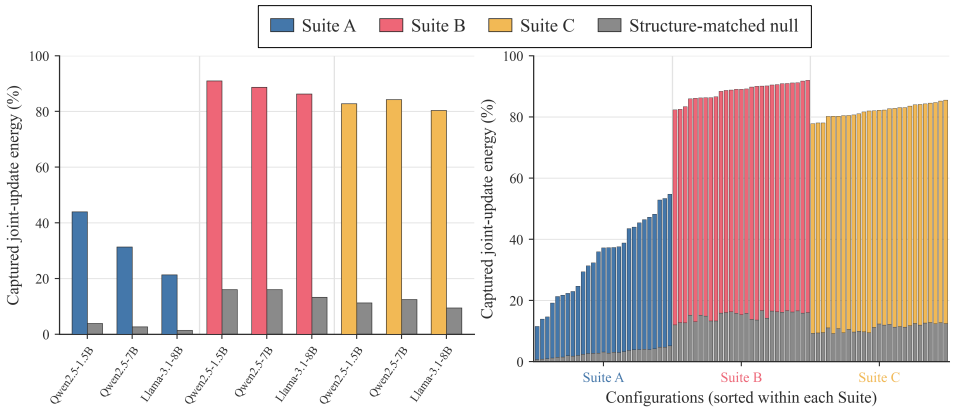
2.1 Even Optimal Endpoint Reweighting Cannot Match Joint Updates

The shortfall of the best reweighting is exactly the part of the joint update orthogonal to the endpoint span. Starting from a shared model θ_0, independent training on task i yields the update $\Delta_i = \theta_i^I - \theta_0$. Stacking the K updates of a composition as $D_K = [\Delta_1, \ldots, \Delta_K]$, the set reachable by unconstrained reweighting is exactly the column space $\mathcal{S}_K = \mathrm{col}(D_K)$, with coefficients $a \in \mathbb{R}^K$ acting on intact task updates under no sign or simplex constraint. The jointly trained update $\Delta_{J,K} = \theta_{J,K} - \theta_0$ splits against this subspace into a part the endpoints can reproduce, $P_K \Delta_{J,K}$ with $P_K = D_K D_K^\dagger$ the orthogonal projector and $\dagger$ the Moore–Penrose pseudoinverse, and an orthogonal residual $C_{\perp,K} = (I - P_K)\Delta_{J,K}$ that no coefficients can touch. Writing $A_K = \|P_K \Delta_{J,K}\|_2^2$ and $B_K = \|C_{\perp,K}\|_2^2$ for the captured and missing energy, orthogonality gives $A_K + B_K = \|\Delta_{J,K}\|_2^2$, and the missing fraction is

$$q_K = \frac{B_K}{A_K + B_K} = \frac{\min_{a \in \mathbb{R}^K} \|\Delta_{J,K} - D_K a\|_2^2}{\|\Delta_{J,K}\|_2^2}. \tag{1}$$

Table 1: Endpoint-span coverage and functional insufficiency across the full 81-cell LLM grid. Values are medians over three LoRA ranks and three seeds; brackets report the interquartile range (IQR). Captured energy is the complementary fraction $1 - q$; the last column counts configurations in which even the oracle in-span composition has higher loss than joint training.

Suite	Backbone	n	Outside energy q	Captured energy	SpanGap	SpanGap > 0
A	Llama-3.1-8B	9	0.787 [0.641, 0.854]	21.3%	0.141 [0.122, 0.150]	9/9
A	Qwen2.5-7B	9	0.687 [0.546, 0.754]	31.3%	0.045 [0.042, 0.056]	9/9
A	Qwen2.5-1.5B	9	0.560 [0.471, 0.612]	44.0%	0.099 [0.096, 0.125]	9/9
B	Llama-3.1-8B	9	0.137 [0.102, 0.167]	86.3%	0.067 [0.049, 0.088]	9/9
B	Qwen2.5-7B	9	0.113 [0.099, 0.137]	88.7%	0.075 [0.050, 0.083]	9/9
B	Qwen2.5-1.5B	9	0.091 [0.088, 0.108]	90.9%	0.041 [0.007, 0.058]	8/9
C	Llama-3.1-8B	9	0.196 [0.190, 0.219]	80.4%	0.096 [0.083, 0.120]	9/9
C	Qwen2.5-7B	9	0.158 [0.153, 0.173]	84.2%	0.064 [0.060, 0.075]	9/9
C	Qwen2.5-1.5B	9	0.172 [0.169, 0.195]	82.8%	0.044 [0.040, 0.063]	9/9
All	All backbones	81	0.178 [0.137, 0.546]	82.2%	0.070 [0.049, 0.098]	80/81



(a) Suite–backbone coverage distributions (b) Percentile profiles versus nulls

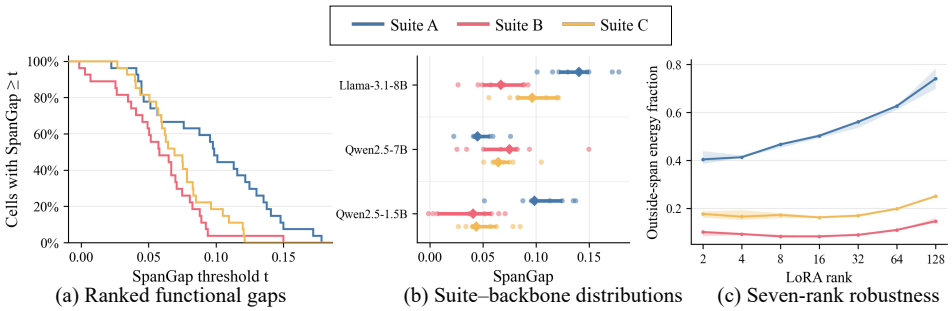
Figure 1: **Endpoint spans capture structured joint-update energy.** (a) Endpoint spans capture substantial joint-update energy across suites and backbones, although the captured fraction varies systematically across groups. (b) Real endpoint spans consistently outperform matched nulls, showing that their alignment with the joint update cannot be explained by dimension alone.

Here q_K is the share of the joint update that even the optimal coefficients leave unreconstructed, and any $q_K > 0$ certifies a direction absent from $\mathcal{S}_K$.

Real joint training leaves a substantial component outside the span, while purely additive targets leave none. An additive-target check confirms the measurement carries no built-in bias: targets $\Delta_{\mathrm{add},K} = D_K a$ reconstruct to numerical zero across every paired comparison in the controlled testbed. The jointly trained updates over the same endpoints behave oppositely, retaining positive outside-span energy in every case; for full five-task compositions, median $q_5 = 0.2253$, placing roughly a fifth of the joint update beyond any endpoint reweighting. Figure 1 confirms this on the full LLM grid: endpoint spans capture structured joint-update energy, but their coverage varies systematically across suites and backbones, yet remains far above matched null subspaces.

2.2 Endpoints Are Geometrically Aligned but Functionally Insufficient

The endpoint span is strongly aligned yet still incomplete, even when enriched. A residual on its own does not say whether the endpoints form a good or poor basis: a Haar-random subspace of matched dimension captures essentially none of the joint update ($q_{\mathrm{Haar}} \approx 0.99999999$), whereas the real spans capture a median of 82.2% across the grid (Table 1). We further test richer endpoint-



(a) Ranked functional gaps (b) Suite–backbone distributions (c) Seven-rank robustness

Figure 2: **Endpoint spans remain functionally incomplete across the full grid.** (a) Joint training outperforms oracle in-span composition in 80 of 81 configurations. (b) The gap persists across suites and backbones. (c) Endpoint-span insufficiency persists across LoRA ranks.

derived oracle spaces motivated by TSV (Gargiulo et al., 2025), AIM (Heyrani Nobari et al., 2026), and OSRM (Zhang & Zhou, 2025). Across the same 81 configurations, all retain nonzero outside-space energy: full-component TSV-style spans and an OSRM-style row–column closure improve every cell, reducing median q from 0.178 to 0.170 and 0.113, while AIM-style relaxations and top-10 TSV-style spans do not improve the endpoint oracle. Even the strongest closure leaves normalized $C_\perp$ residual 0.609 (Appendix B.1).

Yet this aligned representation stays functionally insufficient: optimizing coefficients against the joint loss itself does not close the gap. Reconstruction energy need not track task performance, since different parameters can behave alike, so the geometric residual might in principle be functionally harmless. To rule this out we grant endpoint composition direct access to the joint objective $\mathcal{L}$ and optimize the unconstrained coefficients from multiple starts, allowing a functionally useful combination to depart from the closest parameter reconstruction. For the best coefficients $\widehat{a}_K$, the functional gap is $\mathrm{SpanGap}_K = \mathcal{L}(\theta_0 + D_K\widehat{a}_K) - \mathcal{L}(\theta_0 + \Delta_{J,K})$. Joint training attains the lower loss in 80 of the 81 configurations (Figure 2), so even objective-aware reweighting cannot match it.

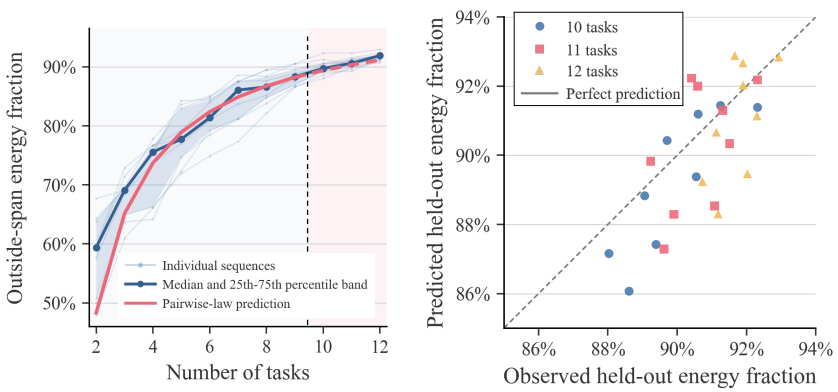
Together these results fix the standing of endpoint reweighting as a useful but incomplete basis: it captures most of the joint update, yet leaves both an exact reconstruction residual and a performance gap that direct optimization does not remove. The tension sharpens as compositions grow, since each added task contributes one endpoint direction but many new joint interactions. Section 3 models how the captured and missing energies accumulate and traces the residual to the joint training path.

3 Explaining the Endpoint Gap: How It Grows and Where It Comes From

Merging more task models widens the gap rather than closing it, and it widens along a predictable law. More endpoints should make the joint update easier to reconstruct, yet the reverse holds: along a fixed nested sequence the captured share falls from 36.1% at two tasks to 7.1% at twelve. We show this decline is not incidental but lawful, governed by a single coefficient that predicts the gap at unseen task counts to within a tenth of a percentage point, and we trace the missing component to the joint training path itself rather than to a poor choice of endpoint weights. The first result makes the gap forecastable; the second explains why no reweighting can remove it.

3.1 Endpoints Capture a Shrinking Share as More Tasks Are Merged

The reconstruction target moves with the composition, so more endpoints cover less. A larger reconstruction space can only help when the target is fixed, but here it is not: each task count merges a different set of models, so the joint update under reconstruction changes with the composition and larger merges face harder targets. Along our fixed nested twelve-task sequence the captured share falls from 36.1% at two tasks to 7.1% at twelve. This single nested trajectory is not in tension with the 82.2% median of Section 2, which averages over an 81-cell grid dominated by smaller, better-



(a) Replicated trajectories (b) Predicted versus observed

Figure 3: **Held-out task-count prediction.** (a) The pairwise law captures the observed scaling and extrapolates beyond the calibration range. (b) Predicted versus observed outside-span energy fraction for the 27 held-out observations at $K = 10, 11, 12$.

aligned compositions; the two summarize different populations. The representation grows while its coverage of its own moving target contracts.

Captured energy scales with tasks, missing energy with pairs. The contraction follows from simple counting: K endpoints stand against $\binom{K}{2}$ task pairs, so if captured energy accrues per task ($A_K \approx aK$) and missing energy per pair ($B_K \approx b\binom{K}{2}$), their ratio grows in proportion to the pairs available per task. We never measure A_K and B_K separately; the observable is their ratio,

$$\frac{q_K}{1-q_K} = \frac{B_K}{A_K} \approx \gamma(K-1), \qquad \gamma = \frac{b}{2a}, \tag{2}$$

equivalently a captured fraction $1-q_K \approx [1+\gamma(K-1)]^{-1}$ that bends toward zero as pairs outnumber tasks, even while the span keeps absorbing energy in absolute terms. This linear-versus-quadratic split is a hypothesis about accumulation, not a separate measurement of each energy; what the data can falsify is the shape of the ratio it implies. The approximation holds when average contribution energies are stable and cross-component inner products do not dominate after projection, conditions derived in the appendix, and larger compositions test the predicted shape directly.

3.2 Observations on Few-Task Merges Predict the Gap at More Tasks

One coefficient from small merges fixes the gap at large merges. Because the accumulation law of Eq. 2 carries a single free parameter γ, the entire trajectory of the gap is pinned down once that parameter is known, and γ is exactly the quantity that small, cheap-to-run merges already reveal. This converts the qualitative observation that coverage shrinks into a quantitative out-of-sample forecast: estimate γ on small compositions, freeze it, and read off every larger task count without any further fitting,

$$\widehat{q}_K = \frac{\widehat{\gamma}(K-1)}{1+\widehat{\gamma}(K-1)}. \tag{3}$$

We compare against a plateau baseline (last-value) and a constant-increment baseline (linear-q). Calibrating on $K = 2, 3$ and predicting $K = 4, 5$ across the controlled testbed, the accumulation model attains the lowest pooled error and beats both baselines on the majority of configurations (Appendix Table 15); its margin over the growth-projecting linear baseline is the informative one, since it shows the gain comes from the shape of the curve rather than the fact of growth.

A coefficient fit on at most nine tasks predicts 92.84% missing energy at twelve, against a measured 92.93%. The longer LLM sequence extends the test to twelve tasks. Fixing $\widehat{\gamma} = 1.178435$ from task counts up to nine and holding it for ten through twelve, the held-out prediction lands within 0.09 percentage points of the measurement at twelve tasks (Figure 3), an accuracy the plateau and

constant-increment baselines miss by wide margins. The gap is therefore not merely monotone: its magnitude is set in advance by a single number read off the smaller merges.

3.3 The Gap Arises Along Training, Not from Suboptimal Weighting

The missing component survives the best reweighting at every step of the joint path. Forecasting the gap leaves its cause open. A fixed endpoint mixture could fail either because the ideal mixture of isolated directions shifts across training, which is coefficient drift, or because joint optimization moves along directions no local mixture can express, which is local escape. Letting the coefficients vary at every interval removes drift by construction, so whatever residual survives is a direction the isolated updates simply cannot supply.

An exact decomposition separates drift from escape. Assigning the residual to these two causes is only meaningful if the split leaves no slack, so that the verdict reflects the trajectory itself rather than an approximation error absorbed into one term. We obtain such a split by comparing, interval by interval, each joint increment with the best reconstruction the isolated updates can offer at that point, and then measuring how the local optima differ from the single final mixture. For a training partition Π, let u_m^J be the joint increment in interval m and U_m the corresponding isolated increments; the best local fit $a_m = U_m^\dagger u_m^J$ leaves $r_m = u_m^J - U_m a_m$. Against the optimal final coefficients $\beta = D_K^\dagger \Delta_{J,K}$, the residual splits as

$$C_{\perp,K} = \underbrace{\sum_m r_m}_{R_\Pi} + \underbrace{\sum_m U_m(a_m - \beta)}_{E_\Pi}, \tag{4}$$

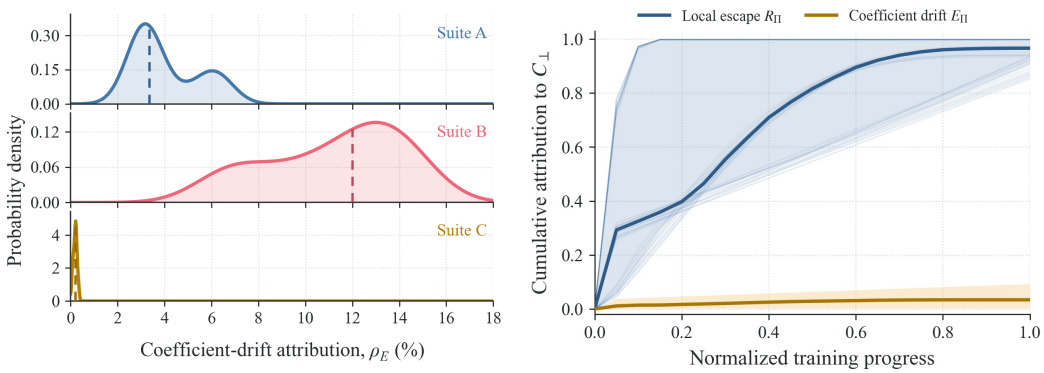
where R_Π accumulates locally unreconstructible updates and E_Π records the cost of replacing interval-specific coefficients with one final mixture. The identity is exact because interval increments sum to their endpoints, and it adjudicates the two causes directly: a dominant E_Π would blame changing mixtures, whereas a dominant R_Π locates the source in locally unavailable updates. The construction and proof are in the appendix.

Local escape, not coefficient drift, dominates the missing direction. Across the audited grid the missing component aligns almost entirely with the locally unreconstructible term (Figure 4), leaving coefficient drift a minor contributor. We measure each contribution by its signed projection along the final residual, $\rho_R = \langle R_\Pi, C_{\perp,K}\rangle / \|C_{\perp,K}\|_2^2$ and ρ_E analogously with $\rho_R + \rho_E = 1$: the median ρ_R is 0.966 and never falls below 0.851 across the configurations passing the numerical audit, with the two audit failures retained as marked review cases. This attribution is geometric, fixing the *direction* of the missing component in parameter space rather than the function it costs, which Section 4 takes up. Because escape dominates even after every interval receives its best possible local coefficients, the missing component is already forming during joint training, before the endpoints are ever combined.

This identifies the path origin of the reconstruction limit in Section 2: the gap grows lawfully with task count and raises from locally unavailable updates along the joint path, a direction endpoint reweighting cannot create. Whether restoring this component recovers the lost function is a separate, functional claim, tested in Section 4 by reinjecting R_Π and measuring the SpanGap it closes.

4 Reinjecting the Missing Direction Improves Model Merging

Reinjection restores function by changing the error that remains after merging. Section 3.3 separates the endpoint residual into accumulated local escape and coefficient drift. Returning R_Π removes the first contribution from the reconstruction error. How much performance returns depends on the functional cost of what remains. This connects the recovery, direction, and dose experiments: each changes a specific aspect of the error presented to the composition objective.



(a) Coefficient-drift distribution

(b) Accumulation along training

Figure 4: **The endpoint-external component is created along the joint path.** (a) The coefficient-drift share ρ_E remains small across all three suites; because $\rho_R = 1 - \rho_E$, most of the endpoint-external component is attributable to local escape. (b) These local-escape contributions accumulate throughout training across 79 audited trajectories.

4.1 Reinjecting the Joint-Path Residual Recovers Function, and the Recovery Is Direction-Specific

Reinjection trades the full endpoint omission for only its drift remainder. The endpoint model $\theta_E = \theta_0 + P_K \Delta_{J,K}$ discards the entire residual $C_{\perp,K}$; adding back R_Π cancels its accumulated local-path part and leaves the coefficient-drift term alone,

$$\theta_E + R_\Pi = \theta_{J,K} - C_{\perp,K} + R_\Pi = \theta_{J,K} - E_\Pi, \tag{5}$$

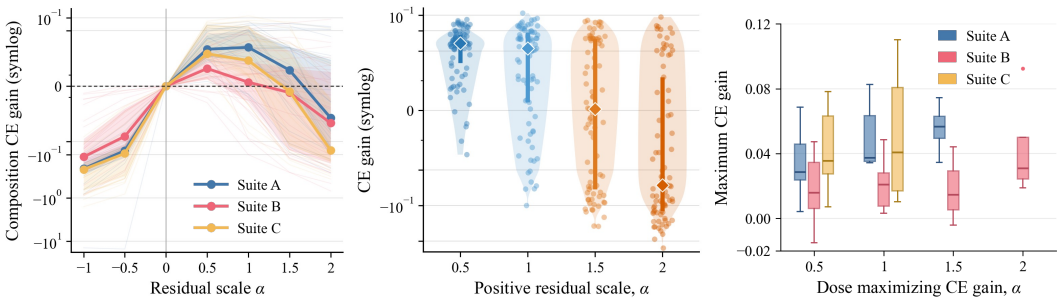
where $\theta_{J,K} = \theta_0 + \Delta_{J,K}$ is the jointly trained model and E_Π is the coefficient-drift term of Section 3.3. The functional question is therefore sharp—is the drift remainder cheaper to omit than the whole residual? The answer is yes: returning R_Π at coefficient $+1$ lowers composition cross-entropy (CE) relative to E in all 78 configurations passing the numerical identity audit, with median paired reduction 0.0815 (Table 10).

A first control asks whether the missing ingredient is simply unused isolated history rather than the joint path itself. The enriched reference $\theta_I = \theta_0 + P_{\text{priv}} \Delta_{J,K}$ projects the joint target onto the endpoints together with every stored isolated-trajectory increment, a strictly larger geometric space than E's. It does not close the gap: reinjection still beats I in 76 of 78 audited configurations (median 0.0803), and, more tellingly, I itself improves on E in only 35 of 81 configurations, with median gain -0.0027. More independent history, optimally projected, buys almost nothing.

This contrast shows that lower parameter error need not imply lower composition loss. Since the private-history span contains the endpoint span, I cannot have larger optimal squared parameter error than E; geometrically it is never worse, but functionally it is a wash. Orthogonal projection prices errors by squared parameter size, whereas composition loss prices them by predictive effect: equal-sized errors can have very different functional costs, so adding directions the loss barely feels leaves CE unchanged. Reinjection helps because it restores the CE-relevant local-path component.

A second control reverses the correction. Equation 5 shows the positive correction leaves E_Π, whereas the sign-reversed control $(E - R_\Pi)$ leaves $2R_\Pi + E_\Pi$, a second copy of the local-path omission already in E. Reversal should thus be worse than leaving the endpoint untouched, and it is by a wide margin: positive reinjection wins in all 81 paired configurations, with median CE advantage 0.3214, the largest contrast in Table 10.

Matched update energy is not enough; the direction has to be right. A third control replaces R_Π with a random vector of the same length. The randomized reference $E + R_{\text{rand}}$ with $\|R_{\text{rand}}\|_2 = \|R_\Pi\|_2$ injects identical energy in an arbitrary orientation, and it fails to recover: positive reinjection wins in all 81 configurations, with median advantage 0.0820. Because the sign and randomized controls never invoke the reconstruction identity of Equation 5, they run on the full 81-configuration



(a) All 567 dose observations (b) Positive-dose distributions (c) Cell-wise best dose and gain

Figure 5: **A useful direction still requires a calibrated scale.** (a) All 81 seven-dose trajectories and suite summaries. (b) All 324 positive-dose observations with medians, interquartile ranges, and improved-cell counts. (c) The best tested positive dose and attained gain for every cell: 43, 27, 7, and 4 cells select $\alpha = 0.5, 1, 1.5, 2$, respectively; four cells obtain no positive tested gain. Positive gain means lower CE than the fixed zero-dose reference.

grid rather than the audited 78. Matched magnitude is therefore insufficient: the recovery lives in the orientation of R_Π, not its length.

4.2 What Merging Should Preserve: A Reusable Direction, Not a Fixed Scale

A useful orientation does not prescribe a useful step size. The magnitude worth applying after reinjection is set by the loss response at the reconstructed model, not by the norm the residual accumulated along its source trajectory: the direction comes from the trajectory, the scale from the loss. This holds even at θ_E, where a unit dose exactly cancels the accumulated R_Π yet still leaves the drift remainder E_Π; exactly canceling one geometric contribution need not minimize the loss while the other remains. The residual norm thus records what accumulated, not how far to move at the receiving model. To measure how far one should move, a separate dose sweep fixes the reference model θ_{ref} and the extracted direction within each configuration and varies $\theta(\alpha) = \theta_{\text{ref}} + \alpha R_\Pi$, with gains measured against this sweep's own zero-dose reference, separately from the geometric recovery experiment above.

A local loss expansion makes the source of scale sensitivity explicit. Let $\phi(\alpha) = \text{CE}(\theta(\alpha))$. Where this function is twice differentiable, its initial slope $s = \phi'(0)$ and curvature $h = \phi''(0)$ give

$$\phi(\alpha) - \phi(0) = s\alpha + \frac{1}{2}h\alpha^2 + o(\alpha^2). \tag{6}$$

The slope s decides whether a small positive step helps at all, while the curvature h decides when the quadratic penalty overtakes that initial benefit. The local quadratic model prefers $\alpha_{\text{quad}} = -s/h$ at the application point, so the residual's accumulated norm does not determine the preferred scale.

The dose sweep shows benefit followed by overshoot: $\alpha = 0.5$ improves $76/81$ configurations with median gain 0.0283, whereas $\alpha = 2.0$ improves only $24/81$ and has median gain -0.0401 (Figure 5; Appendix Table 11). The best scale is configuration-dependent: $\alpha = 0.5$ is best in 43 cells, while 38 prefer a larger dose. Thus the pairwise law predicts how much residual accumulates, while the dose sweep shows that applying the recovered direction requires a loss-calibrated scale.

The best intervention scale varies across configurations. Among the tested positive doses $\{0.5, 1.0, 1.5, 2.0\}$, half magnitude is best in 43 configurations; the other 38 favor a larger dose. A common coefficient therefore does not maximize the measured gain across configurations. These differing optima are consistent with different loss responses along the extracted directions. The accumulation model in Section 3 characterizes how much missing energy develops; applying an extracted component also requires knowing how the receiving objective responds to it.

A merging interface needs both a correction direction and a functional rule for applying it. Here, reuse takes a direction from a configuration's joint trajectory and applies it to that configu-

ration's reconstructed model. The resulting design target is to preserve or estimate a useful coordination direction, such as $d_\Pi = R_\Pi / \|R_\Pi\|_2$, and calibrate its applied magnitude on the intended composition objective. Joint training supplies the reference direction in these experiments. The broader implication is that a training record becomes useful for composition through a readout that selects which error to correct and how far to move.

5 Related Work

Endpoint-based model merging. Model merging composes independently fine-tuned checkpoints into a multi-task model. Task Arithmetic represents each task as a displacement from a shared pretrained model and composes capabilities by vector addition or negation (Ilharco et al., 2023); TIES-Merging trims small updates and resolves sign conflicts (Yadav et al., 2023), DARE drops and rescales delta parameters to exploit redundancy (Yu et al., 2024), and AdaMerging learns task- or layer-wise coefficients from unlabeled test data (Yang et al., 2024b). These endpoint interfaces remain bounded by what isolated training encodes (Yang et al., 2026a), including recent variants using task singular vectors, activation-informed importance, or orthogonal LoRA subspaces (Gargiulo et al., 2025; Heyrani Nobari et al., 2026; Zhang & Zhou, 2025)

Mechanisms and limits of endpoint composability. Work on endpoint composability explains when such operations succeed, through permutation alignment (Ainsworth et al., 2022), tangent-space disentanglement (Ortiz-Jimenez et al., 2023), submodule linearity (Dai et al., 2025), conditional task-vector guarantees (Li et al., 2025), singular-subspace preservation or confidence-aware scaling (Sun et al., 2026; Yoshida et al., 2026), and merge-robust fine-tuning (Lee et al., 2025; Yang et al., 2026b). Other analyses connect merge degradation to optimizer noise, expert overtraining, and task count (Zhang et al., 2026; Horoi et al., 2025; Wang et al., 2026; 2025; Song & Zheng, 2026; Yang et al., 2026a), with one two-expert RL study reporting parity in a near-orthogonal regime (McClendon, 2026). These accounts stay inside the endpoint span; what joint training accumulates outside it remains unaddressed.

Task interaction during joint optimization. Multi-task optimization typically reads task interaction off gradients at individual training states: TAG uses inter-task loss affinities for grouping (Fifty et al., 2021), PCGrad projects away conflicting gradient components (Yu et al., 2020), and CAGrad balances descent on the average objective against the worst local conflict (Liu et al., 2021). Beyond gradient manipulation, CORE-MTL traces negative transfer to representation entanglement via causal orthogonal representations (Wu et al., 2026), and Mueller et al. compare single- and multi-task trajectories, finding that gradient conflict shapes optimization but does not reliably predict generalization (Mueller et al., 2025). These accounts capture interaction at local states, missing what the joint trajectory accumulates over its full path.

6 Conclusion and Discussion

Model merging assumes that reweighting independently trained endpoints can stand in for joint training. We show that this assumption fails in a structural but predictable way: endpoint spans capture most of the joint update, yet miss a direction no reweighting can express. The missing energy follows a one-parameter pairwise law that predicts held-out task counts, and the direction itself arises as a path-accumulated residual of joint optimization. Reinjection restores its functional effect, while sign-reversed and norm-matched controls do not. Joint training therefore contributes not only endpoints, but a coordination direction that independent runs fail to record.

Implications for merging. Our oracle reinjection does not propose a practical estimator; it identifies what such an estimator must recover. The target separates into a direction created by co-learning and a scale set by the receiving objective. Since the magnitude is predictable from small merges, future methods can focus on estimating the direction from short joint segments or pairwise probes, then calibrating its dose against the downstream loss.

Scope and generalization. Our evidence covers 81 LLM fine-tuning configurations, a controlled multi-task testbed, and compositions of up to twelve tasks under low-rank adaptation. Full fine-tuning, larger task counts, nonlinear merging interfaces, and other modalities remain open, but the pairwise law and path-residual decomposition give concrete tests for those settings.

AI USE STATEMENT

In this work, we used generative AI tools solely to improve the grammar, clarity, and readability of the manuscript. Generative AI tools were not used to generate research ideas, data, results, analyses, conclusions, or references, and other disclosure categories are not applicable to this work. All AI-assisted revisions were reviewed and edited by the authors to ensure their accuracy and consistency with the original scientific meaning. We take full responsibility for the final content of this work.

REFERENCES

Samuel K Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. *arXiv preprint arXiv:2209.04836*, 2022.

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*, volume 2025, pp. 10581–10608, 2025.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34:27503–27516, 2021.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodola. Task singular vectors: Reducing task interference in model merging. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18695–18705. IEEE, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Amin Heyrani Nobari, Kaveh Alimohammadi, Ali ArjomandBigdeli, Akash Srivastava, Faez Ahmed, and Navid Azizan. Activation-informed merging of large language models. *Advances in neural information processing systems*, 38:72975–72998, 2026.

Stefan Horoi, Guy Wolf, Eugene Belilovsky, and Gintare Karolina Dziugaite. From memorization to parameter interference: How overtraining experts harms model merging. *arXiv preprint arXiv:2506.14126*, 2025.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Yeoreum Lee, Jinwook Jung, and Sungyong Baik. Mitigating parameter interference in model merging via sharpness-aware fine-tuning. In *International Conference on Learning Representations*, volume 2025, pp. 50156–50178, 2025.

Hongkang Li, Yihua Zhang, Pin-Yu Chen, Sijia Liu, Meng Wang, et al. When is task vector provably effective for model editing? a generalization analysis of nonlinear transformers. In *International Conference on Learning Representations*, volume 2025, pp. 101273–101305, 2025.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

S Aaron McClendon. When model merging rivals joint multi-task reinforcement learning: A task-vector geometry analysis. *arXiv preprint arXiv:2607.16062*, 2026.

David Mueller, Mark Dredze, and Nicholas Andrews. Can optimization trajectories explain multi-task transfer? *Transactions on Machine Learning Research*, 2025.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36:66727–66754, 2023.

Mingyang Song and Mao Zheng. Model merging in the era of large language models: Methods, applications, and future directions. *arXiv preprint arXiv:2603.09938*, 2026.

Wenju Sun, Qingyong Li, Tiancheng Li, Yangliao Geng, and Boyang Li. Revisiting the role of pretrained weights in model merging: On near-optimality within the core subspace. In *Forty-third International Conference on Machine Learning*, 2026.

Yuanyi Wang, Yanggan Gu, Yiming Zhang, Qi Zhou, Zhaoyi Yan, Congkai Xie, Xinyao Wang, Jianbo Yuan, and Hongxia Yang. Model merging scaling laws in large language models. *arXiv preprint arXiv:2509.24244*, 2025.

Zijing Wang, Xingle Xu, Yongkang Liu, Yiqun Zhang, Peiqin Lin, Shi Feng, Daling Wang, Xiaocui Yang, and Hinrich Schütze. Why do more experts fail? a theoretical analysis of model merging. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45460–45482, 2026.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022.

Chengfeng Wu, Tao Zou, Yanru Wu, and Jingge Wang. CORE-MTL: Rethinking gradient balancing via causal orthogonal representations. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=vIIk6uur7Z.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024a. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, 2024b.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities. *ACM Computing Surveys*, 58(8):1–41, 2026a.

Enneng Yang, Qun Yang, Peng Wang, Anke Tang, Guibing Guo, Xiaochun Cao, and Li Shen. Mergopt: A merge-aware optimizer for robust model merging. In *International Conference on Learning Representations*, volume 2026, pp. 48400–48425, 2026b.

Kotaro Yoshida, Yuji Naraki, Takafumi Horie, Ryotaro Shimizu, Ioannis Mitliagkas, and Hiroki Naganuma. Distac: Conditioning task vectors via distillation for robust model merging. In *International Conference on Learning Representations*, volume 2026, pp. 118509–118534, 2026.

Linyan 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 *Icml*, volume 2, pp. 21, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Chenxiang Zhang, Alexander Theus, Damien Teney, Antonio Orvieto, Jun Pang, and Sjouke Mauw. How does the optimizer implicitly bias the model merging loss landscape? In *International Conference on Learning Representations*, volume 2026, pp. 97540–97574, 2026.

Haobo Zhang and Jiayu Zhou. Unraveling lora interference: Orthogonal subspaces for robust model merging. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26459–26472, 2025.

A Experimental Protocols and Audit Policy

Controlled five-task benchmark. We use a nonlinear teacher–student system with five tasks, task-signal rank R, and shared bottleneck width b. We evaluate the 15 rank–capacity conditions

$$R = 1 :\ b \in \{1, 2, 4\}, \quad R = 2 :\ b \in \{1, 2, 4\},$$
$$R = 3 :\ b \in \{2, 3, 6\}, \quad R = 4 :\ b \in \{3, 4, 6\},$$
$$R = 5 :\ b \in \{4, 5, 8\}.$$

For $R \geq 2$, these include capacity-pressure ($b < R$), matched ($b = R$), and slack ($b > R$) regimes; $R = 1$ uses one matched and two increasingly slack conditions. Repeating every condition under three seeds gives 45 configurations. Within each configuration, all 31 nonempty task subsets are trained for 1,000 optimizer steps from a common initialization. Every active task supplies one minibatch per step, task losses are averaged before a single update, and overlapping subsets use identical per-task minibatch streams. The 26 subsets containing at least two tasks therefore produce $45 \times 26 = 1{,}170$ paired comparisons. PARAM-ADD sums the corresponding isolated parameter updates, whereas JOINT is obtained by training directly on the multitask objective.

LLM tasks and splits. All tasks are formulated as multiple-choice answer-letter prediction using deterministic, disjoint train/calibration/evaluation splits. Suite A contains AQuA-RAT, BoolQ, HellaSwag, and PIQA, each with $6000/1000/1000$ examples. Suite B contains the MMLU-Pro history ($269/39/39$), law ($764/109/111$), biology ($488/69/72$), and mathematics ($946/136/136$) subjects. Suite C uses P3 template 0 for AG News ($718/82/800$), IMDb ($718/81/798$), PAWS ($711/89/800$), and ANLI-R1 ($716/83/800$). Each triple denotes train/calibration/evaluation counts after preprocessing. We discard examples whose prompt exceeds 1,024 tokens under any of the three backbone tokenizers; the supervised target is the next answer-letter token.

Paired LLM training. The main grid contains 3 task suites $\times$ 3 backbones $\times$ 3 LoRA ranks $\times$ 3 seeds $= 81$ cells. The backbones are Llama-3.1-8B, Qwen2.5-7B, and Qwen2.5-1.5B; the LoRA ranks are $r \in \{8, 32, 64\}$ and the seeds are $\{30, 36, 42\}$. The backbone is frozen and LoRA is applied only to the q_proj and v_proj modules, with $\alpha = 2r$, zero dropout, and no adapter bias. Within each cell, isolated and joint runs share the adapter initialization and task-specific example order. We train for two epochs with a per-task batch size of 32. The joint schedule runs for the longest task; each task follows the same minibatch stream as its isolated run and contributes with weight $1/4$ while data remain. Optimization uses AdamW with learning rate 2×10^{-4}, $\beta = (0.9, 0.95)$, zero weight decay, 3% warmup, and cosine decay. Training minimizes the full-vocabulary next-token negative log-likelihood of the answer letter.

Geometry and functional evaluation. For each LoRA module, we form its effective dense update $\Delta W = (\alpha/r)BA$ and sum Frobenius inner products across all adapted modules. Endpoint projections are computed from float64 Gram systems using the Moore–Penrose pseudoinverse with $\texttt{rcond} = 10^{-10}$. The functional oracle imposes no sign or simplex constraint: coefficients are selected on the calibration NLL from the projection and all-ones initializations and are then evaluated on the held-out evaluation split. Reported task losses are macro-averaged across the four tasks.

For trajectory analysis, we save the initialization and steps $1, 2, 5, 10$, every 25 steps thereafter, and the final step. Each interval is projected onto its contemporaneous isolated-increment span. Coarse partitions with $G \in \{2, 4, 8\}$ are constructed by first summing the underlying increments and then recomputing the projection; fine residuals are never summed as a substitute. We additionally repeat the small-system solve at rcond $\in \{10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}\}$.

Metric conventions. The endpoint residual fraction q is the fraction of joint-update energy outside the relevant span, so smaller values indicate greater coverage. $\mathrm{SpanGap} > 0$ means that the jointly trained model has lower held-out NLL than the best endpoint combination. For enlarged-subspace stress tests we report the paired reduction $\Delta q = q_{\mathrm{endpoint}} - q_{\mathrm{method}}$, so positive values favor the enlarged subspace. Alignment and signed recovery retain their algebraic signs.

Numerical audits. All 1,170 PARAM-ADD targets reconstruct to numerical zero, whereas every paired JOINT target has positive outside-span energy. Sign and matched-null checks cover all 81 LLM cells. The recovery-identity analysis contains 78 fully identity-audited configurations. The exact path decomposition received an effective PASS in 79 of 81 cells; the remaining two are marked REVIEW and are retained transparently rather than silently discarded.

Implementation. LLM experiments use Python 3.10, PyTorch 2.8.0 with CUDA 12.8, Transformers 5.9.0, and PEFT 0.19.1 on NVIDIA H200 GPUs. Backbones are loaded in bfloat16, while Gram matrices and pseudoinverse calculations use float64.

B Endpoint Sufficiency: Complete Results and Robustness

Controlled additive calibration. For $\Delta_{\mathrm{add},K} = D_K a$, projection onto $\mathrm{col}(D_K)$ must leave zero residual. The controlled benchmark contains 45 configurations and 1,170 paired subset comparisons. PARAM-ADD reaches numerical zero in 1,170/1,170 comparisons; JOINT has positive outside energy in 1,170/1,170. For complete five-task compositions, median $q_5 = 0.2253$.

Table 2: Matched calibration of the projection measurement.

Target	Positive outside energy	Complete-K summary
PARAM-ADD	$0/1,170$	numerical zero
JOINT	1,170/1,170	median $q_5 = 0.2253$

B.1 Advanced Endpoint-Derived Representational Stress Tests

Protocol. We test whether the endpoint-external component remains absent after enriching the isolated-run representation with structures motivated by TSV, AIM, and OSRM. For every cell in the 81-configuration grid, the joint update is used only as an oracle reconstruction target. The TSV-style constructions expose either the leading ten or all per-layer singular components of each isolated task update. The AIM-style construction applies activation-derived diagonal relaxation, computed from 32 task-agnostic calibration sequences of length 256, over $\omega \in \{0, 0.25, 0.5, 0.75, 1\}$. The OSRM-style construction forms the per-layer row–column closure generated by the stored isolated LoRA factors.

These constructions are favorable representational stress tests rather than end-to-end reproductions of the original algorithms: OSRM constrains LoRA subspaces before fine-tuning, whereas our experiment asks what the analogous closure can recover post hoc; similarly, the TSV- and AIM-style variants evaluate oracle joint-update reconstruction rather than downstream merging performance.

Let $\widehat{\Delta}_E = P_K \Delta_{J,K}$ be the intact-endpoint oracle and $\widehat{\Delta}_M$ the reconstruction under construction M. We report

$$q_M = \frac{\|\Delta_{J,K} - \widehat{\Delta}_M\|_2^2}{\|\Delta_{J,K}\|_2^2}, \qquad \Delta q_M = q_E - q_M,$$

so positive Δq_M denotes improvement. Defining the additional correction $g_M = \widehat{\Delta}_M - \widehat{\Delta}_E$, alignment is $\cos(g_M, C_{\perp,K})$, signed recovery is $\langle g_M, C_{\perp,K}\rangle / \|C_{\perp,K}\|_2^2$, and the remaining normalized residual is $\|C_{\perp,K} - g_M\|_2^2 / \|C_{\perp,K}\|_2^2$.

The full-component TSV-style span and the row–column closure improve reconstruction in every configuration, confirming that isolated endpoint factor structure contains information about $C_\perp$. Neither construction eliminates the residual: all tested methods retain $q > 10^{-6}$ in all 81 cells, and even the strongest closure leaves median $q = 0.113$ and normalized remaining residual 0.609. For

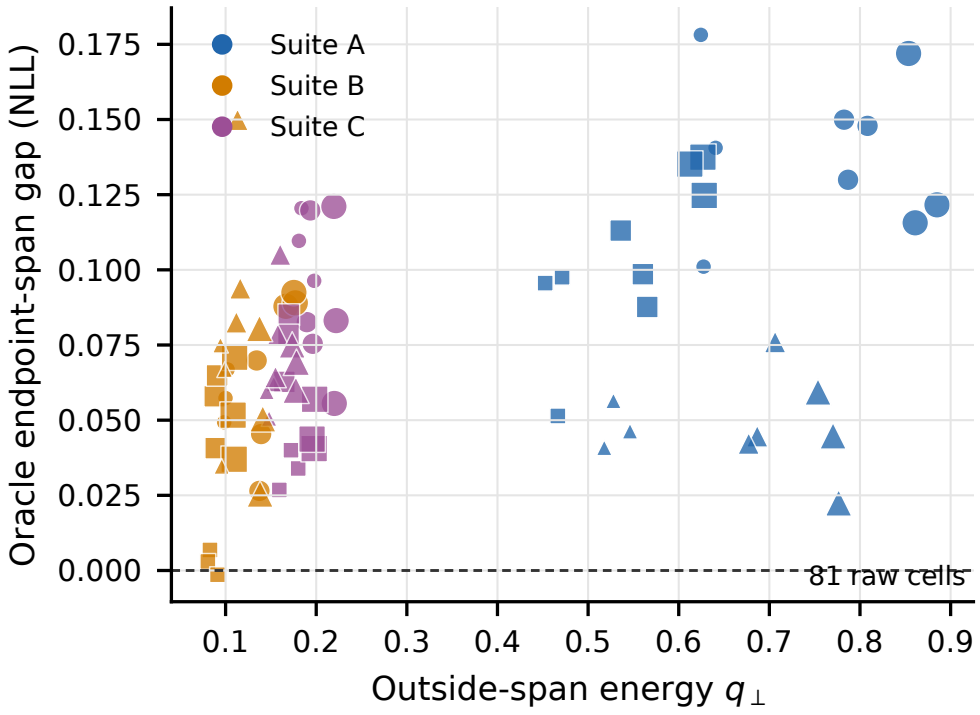


Figure 6: Endpoint geometry across the full LLM grid. The final file exposes all 81 cells; the fallback summarizes the same grid by condition means and seed variability.

Construction	q median [IQR] ↓	Δq [95% CI] ↑	B/T/W	$q > 10^{-6}$	Align.	Rec.	Rem. ↓
Intact endpoint oracle	0.178208 [0.137, 0.546]	—	—	81/81	—	0.000000	1.000000
AIM-style, $\omega = 0.75$	0.178470 [0.137651, 0.546388]	−0.000311 [−0.000357, −0.000242]	0/0/81	81/81	−0.001252	−0.000039	1.000944
TSV-style, top 10 components	0.280913 [0.213497, 0.633739]	−0.064762 [−0.072971, −0.057097]	0/0/81	81/81	0.088489	0.055779	1.265141
TSV-style, all components	0.170011 [0.129781, 0.515163]	+0.011185 [+0.008424, +0.012523]	81/0/0	81/81	0.236578	0.055969	0.944031
OSRM-style row–column closure	**0.112669** [0.074877, 0.270878]	**+0.068715** [+0.063584, +0.094093]	81/0/0	81/81	**0.625080**	**0.390713**	**0.609275**

Table 3: **Advanced endpoint-derived representational stress tests.** $\Delta q = q_{\text{endpoint}} - q_M$, so positive values favor construction M; intervals are 20,000-sample paired-bootstrap 95% confidence intervals. B/T/W counts cells with lower/equal/higher q than the intact-endpoint oracle. Align., Rec., and Rem. denote alignment with $C_\perp$, signed $C_\perp$ recovery, and remaining normalized $C_\perp$ residual, respectively.

AIM, median q is 0.182879, 0.180805, 0.179335, 0.178470, and 0.178208 as ω increases from 0 to 1; $\omega = 1$ recovers the intact-endpoint oracle, and every nontrivial relaxation is worse.

C Task-Count Scaling and Replication

If captured energy accumulates at average rate a per task and missing energy at average rate b per task pair, then

$$A_K \approx aK, \qquad B_K \approx bK(K-1)/2, \qquad \frac{q_K}{1-q_K} \approx \frac{b}{2a}(K-1).$$

The approximation is about the post-projection ratio; it does not assert that all cross terms vanish individually. Each fit uses only its declared calibration counts, and held-out counts are not used to select γ.

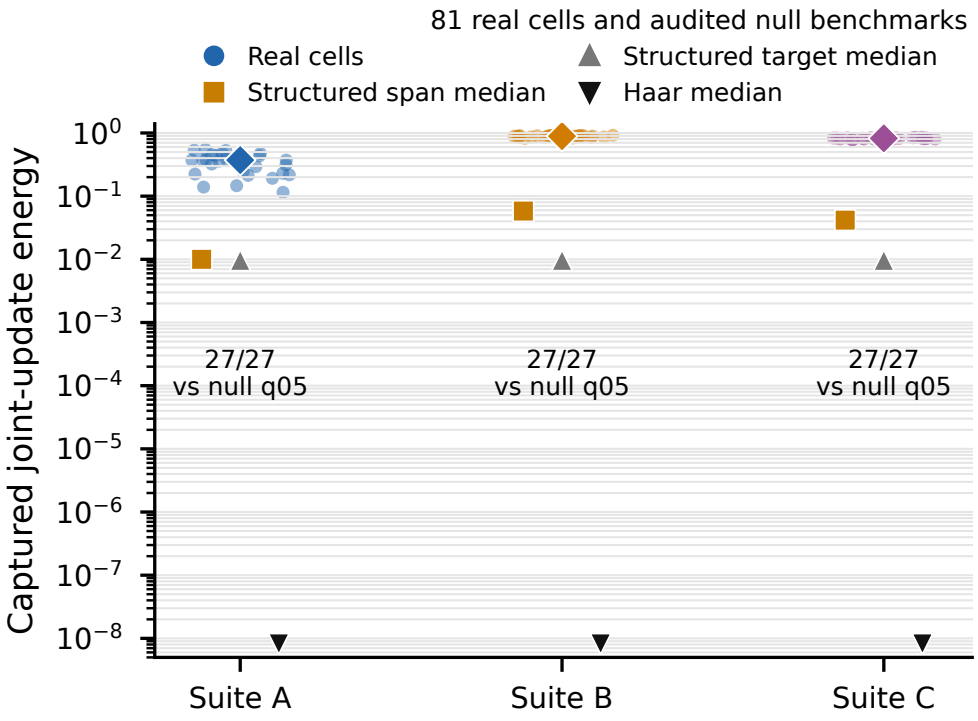


Figure 7: Dimension and construction nulls. The final panel exposes all 81 real cells against suite-level structured-span and global structured-target and Haar benchmarks; every real cell beats its matched structured-null fifth percentile.

D Exact Path Decomposition

For interval m, let u_m^J be the joint increment, U_m the corresponding isolated increments, $a_m = U_m^\dagger u_m^J$, and $r_m = u_m^J - U_m a_m$. If $\beta = D_K^\dagger \Delta_{J,K}$, then

$$\sum_m r_m + \sum_m U_m(a_m - \beta) = \sum_m u_m^J - \sum_m U_m \beta = \Delta_{J,K} - D_K \beta = C_{\perp,K}.$$

This proves $C_{\perp,K} = R_\Pi + E_\Pi$ exactly up to numerical precision. The signed attribution statistics are reported separately from this identity check.

E Functional Recovery, Directional Controls, and Dose

Exact aggregate values for the recovery and directional-control comparisons are retained in Table 10; the main text uses the aligned effect-size and improvement-rate display to avoid repeating the same five numbers in two visual forms.

F Complete Numerical Tables

G Final Audit and Trajectory-Control Tables

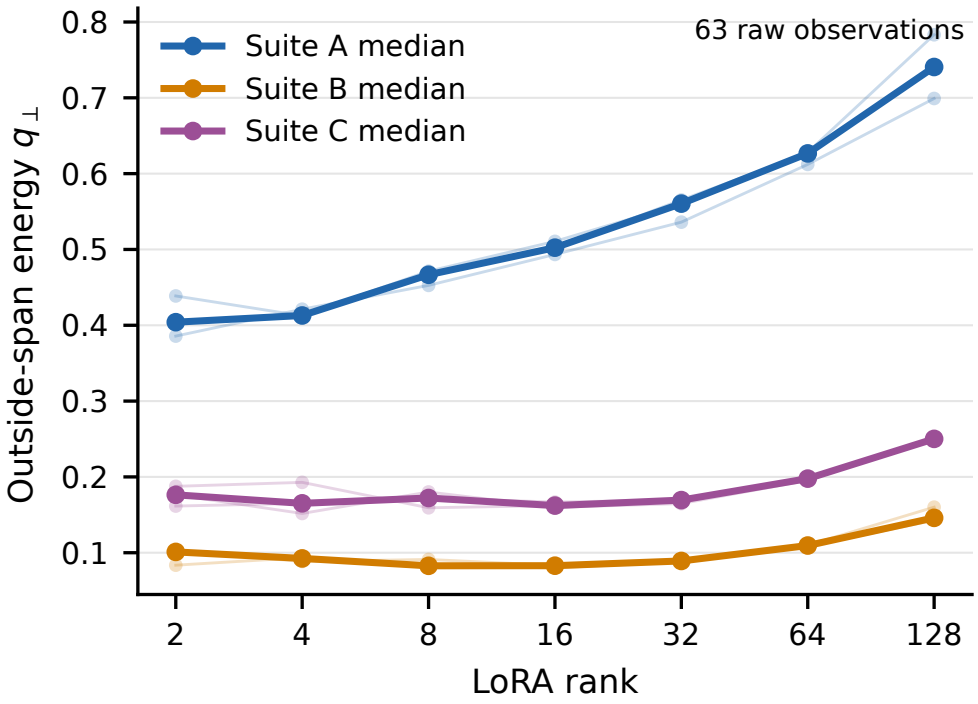


Figure 8: Seven-point LoRA-rank sweep. The final file shows nine matched seed trajectories and all 63 cells; the fallback reports condition means and standard deviations.

Table 4: Complete 27-condition summary of the 3 suites × 3 backbones × 3 ranks grid. Every row is mean±SD over three seeds; the 27 rows summarize 81 cells without hiding the tested conditions. The ρ_R column uses the optimized exact path run; its two numerical-review cells are retained here only for descriptive completeness, while claimed path statistics use the 79 PASS cells.

Suite	Backbone	Rank	$q_\perp$	Δ_{span}	ρ_R	G_{+1}	G_{-1}
A	Llama-3.1-8B	8	0.6308 ± 0.0087	0.1399 ± 0.0385	0.9481 ± 0.0070	0.0526 ± 0.0316	-0.2240 ± 0.0458
A	Llama-3.1-8B	32	0.7925 ± 0.0139	0.1426 ± 0.0110	0.9378 ± 0.0028	0.0456 ± 0.0124	-0.4343 ± 0.3293
A	Llama-3.1-8B	64	0.8665 ± 0.0164	0.1364 ± 0.0309	0.9383 ± 0.0043	0.0307 ± 0.0474	-4.8737 ± 7.9983
A	Qwen2.5-1.5B	8	0.4635 ± 0.0097	0.0815 ± 0.0261	0.9689 ± 0.0011	0.0460 ± 0.0103	-0.1417 ± 0.0066
A	Qwen2.5-1.5B	32	0.5539 ± 0.0157	0.0998 ± 0.0127	0.9696 ± 0.0043	0.0416 ± 0.0224	-0.1879 ± 0.0356
A	Qwen2.5-1.5B	64	0.6223 ± 0.0088	0.1325 ± 0.0068	0.9689 ± 0.0023	0.0407 ± 0.0181	-0.2286 ± 0.0173
A	Qwen2.5-7B	8	0.5307 ± 0.0144	0.0479 ± 0.0079	0.9709 ± 0.0025	0.0354 ± 0.0019	-0.1277 ± 0.0287
A	Qwen2.5-7B	32	0.6901 ± 0.0149	0.0543 ± 0.0188	0.9672 ± 0.0015	0.0339 ± 0.0131	-0.3312 ± 0.1334
A	Qwen2.5-7B	64	0.7668 ± 0.0118	0.0422 ± 0.0186	0.9648 ± 0.0029	0.0070 ± 0.0243	-0.2903 ± 0.3158
B	Llama-3.1-8B	8	0.1002 ± 0.0019	0.0578 ± 0.0088	0.8553 ± 0.0036	0.0129 ± 0.0165	-0.0495 ± 0.0295
B	Llama-3.1-8B	32	0.1369 ± 0.0025	0.0472 ± 0.0217	0.9051 ± 0.0036	0.0081 ± 0.0213	-0.3240 ± 0.1607
B	Llama-3.1-8B	64	0.1730 ± 0.0056	0.0898 ± 0.0024	0.9356 ± 0.0015	-0.0056 ± 0.0478	-0.6533 ± 0.3581
B	Qwen2.5-1.5B	8	0.0848 ± 0.0056	0.0028 ± 0.0041	0.8690 ± 0.0059	0.0142 ± 0.0054	-0.0252 ± 0.0158
B	Qwen2.5-1.5B	32	0.0893 ± 0.0013	0.0545 ± 0.0125	0.8602 ± 0.0097	-0.0250 ± 0.0746	-0.1056 ± 0.0285
B	Qwen2.5-1.5B	64	0.1092 ± 0.0009	0.0532 ± 0.0170	0.8915 ± 0.0030	-0.0255 ± 0.0199	-0.1903 ± 0.0747
B	Qwen2.5-7B	8	0.0963 ± 0.0025	0.0588 ± 0.0213	0.8753 ± 0.0058	-0.0022 ± 0.0115	-0.0100 ± 0.0032
B	Qwen2.5-7B	32	0.1138 ± 0.0022	0.1088 ± 0.0361	0.8721 ± 0.0089	0.0390 ± 0.0172	-0.1071 ± 0.0373
B	Qwen2.5-7B	64	0.1388 ± 0.0019	0.0521 ± 0.0276	0.9245 ± 0.0038	0.0083 ± 0.0226	-0.1561 ± 0.0542
C	Llama-3.1-8B	8	0.1874 ± 0.0090	0.1088 ± 0.0121	0.9990 ± 0.0004	0.0677 ± 0.0114	-0.7248 ± 0.0905
C	Llama-3.1-8B	32	0.1931 ± 0.0033	0.0926 ± 0.0238	0.9986 ± 0.0008	0.0654 ± 0.0201	-1.0729 ± 0.4471
C	Llama-3.1-8B	64	0.2204 ± 0.0013	0.0866 ± 0.0329	0.9981 ± 0.0005	0.0802 ± 0.0262	-0.8626 ± 0.2312
C	Qwen2.5-1.5B	8	0.1705 ± 0.0105	0.0336 ± 0.0066	0.9987 ± 0.0004	0.0130 ± 0.0035	-0.0517 ± 0.0056
C	Qwen2.5-1.5B	32	0.1680 ± 0.0027	0.0754 ± 0.0115	0.9979 ± 0.0004	0.0182 ± 0.0095	-0.1822 ± 0.0321
C	Qwen2.5-1.5B	64	0.1971 ± 0.0015	0.0471 ± 0.0087	0.9976 ± 0.0006	-0.0114 ± 0.0204	-0.3134 ± 0.0368
C	Qwen2.5-7B	8	0.1488 ± 0.0040	0.0574 ± 0.0060	0.9979 ± 0.0002	0.0392 ± 0.0034	-0.0983 ± 0.0137
C	Qwen2.5-7B	32	0.1576 ± 0.0026	0.0826 ± 0.0207	0.9968 ± 0.0003	-0.0160 ± 0.0258	-0.2172 ± 0.0343
C	Qwen2.5-7B	64	0.1764 ± 0.0026	0.0681 ± 0.0076	0.9975 ± 0.0001	0.0005 ± 0.0152	-0.2094 ± 0.0110

Table 5: Seven-point LoRA-rank sweep on Qwen2.5-1.5B. Values are mean±SD over three seeds. $q_\perp(r = 128) > q_\perp(r = 2)$ in all nine matched suite–seed blocks.

Suite	$r = 2$	$r = 4$	$r = 8$	$r = 16$	$r = 32$	$r = 64$	$r = 128$
A	0.4095 ± 0.0269	0.4155 ± 0.0049	0.4635 ± 0.0097	0.5021 ± 0.0086	0.5539 ± 0.0157	0.6223 ± 0.0088	0.7410 ± 0.0420
B	0.0957 ± 0.0105	0.0915 ± 0.0027	0.0848 ± 0.0056	0.0821 ± 0.0017	0.0893 ± 0.0013	0.1092 ± 0.0009	0.1496 ± 0.0092
C	0.1752 ± 0.0131	0.1699 ± 0.0209	0.1705 ± 0.0105	0.1630 ± 0.0035	0.1680 ± 0.0027	0.1971 ± 0.0015	0.2502 ± 0.0006

Table 6: Dense ΔW control, eliminating LoRA parameterization. All nine cells pass the nonzero-residual, positive span-gap, positive-direction, and negative-direction tests.

Backbone	Seed	$q_\perp$	Δ_{span}	G_+	P_-	Joint NLL	Span NLL	λ^*
Qwen2.5-1.5B	30	0.632378	0.086119	0.063150	0.257905	0.690543	0.776662	1.0
Qwen2.5-1.5B	36	0.657636	0.219649	0.134272	0.312189	0.675692	0.895341	1.5
Qwen2.5-1.5B	42	0.630909	0.057783	0.037160	0.154734	0.679437	0.737220	1.0
Qwen2.5-7B	30	0.715989	0.142772	0.105505	0.320052	0.590106	0.732878	1.0
Qwen2.5-7B	36	0.750784	0.152528	0.087707	0.181673	0.636776	0.789304	1.0
Qwen2.5-7B	42	0.749696	0.088885	0.195711	0.461061	0.671287	0.760173	1.0
Llama-3.1-8B	30	0.979417	0.177195	0.179977	0.609102	0.809545	0.986740	0.5
Llama-3.1-8B	36	0.971505	0.811347	0.086287	0.010467	0.781803	1.593150	2.0
Llama-3.1-8B	42	0.970427	0.545363	0.342648	0.500030	0.808845	1.354208	1.0

Table 7: Post-hoc merging baselines. Rows with $n = 78$ are aggregate robustness evidence rather than a perfectly paired tournament against $n = 81$ methods.

Method	n	CE ↓	MergeGap ↓	$\Pr(\text{Gap} > 0)$	JGR ↑	Acc. ↑	WorstCE ↓
Base	81	0.88476	0.28834	1.000	0.000	0.5918	1.60133
Joint training	81	0.57961	0.00000	0.000	1.000	0.7402	1.40527
Task Arithmetic	78	0.68582	0.06788	0.962	0.710	0.7188	1.53566
TIES	78	0.67610	0.07293	0.974	0.745	0.7197	1.50028
Layer-wise AdaMerging (corrected v2)	81	0.94639	0.11400	0.975	0.650	0.7031	1.90856
KnOTS	78	0.66943	0.06922	0.974	0.718	0.7139	1.50292
Core Space	78	0.69517	0.07504	0.949	0.697	0.6973	1.52110
NSC Merging	78	0.69665	0.08659	1.000	0.666	0.7139	1.56369
Expert Merging++ (LoRA port)	78	0.65203	0.06119	0.974	0.800	0.7275	1.51440
Oracle best endpoint span	78	0.66127	0.04925	0.974	0.751	0.7168	1.47972

Table 8: Reference task-count sequence (Qwen2.5-1.5B, rank 8, seed 42). The replicated nine-sequence audit is reported separately in Table 20.

K	$q_\perp$	Δ_{span}	$G^{\text{NLL}}_{+.5}$	$G^{\text{NLL}}_{-.5}$	$C_\perp - \text{random}$
2	0.6386	+0.15794	+0.16470	-0.72319	+0.32001
3	0.6962	+0.09351	+0.12431	-1.15057	+0.12354
4	0.7778	+0.04387	+0.11430	-0.85464	+0.12486
5	0.8293	+0.05565	+0.15586	-1.35563	+0.40077
6	0.8480	+0.04663	+0.11273	-1.49028	+0.31269
7	0.8859	+0.05766	+0.10027	-1.36290	+0.36546
8	0.8838	+0.11637	+0.09084	-0.94130	+0.12768
9	0.9052	+0.14401	+0.10991	-1.29354	+0.22519
10	0.9232	+0.12085	+0.10609	-1.16120	+0.29922
11	0.9232	+0.12119	+0.09630	-1.32988	+0.11531
12	0.9293	+0.13498	+0.11266	-1.16527	+0.16878

Table 9: Endpoint-span geometry versus matched null constructions over 81 cells. Smaller $q_\perp$ means greater capture.

Construction	Median $q_\perp$	Median capture
Real isolated endpoints	0.17820790	0.82179210
Structured span null	0.95826102	0.04173898
Structured target null	0.99036735	0.00963265
Haar dimension null	0.99999999	0.00000001

Table 10: Exact aggregate functional gains for recovery, isolated-history enrichment, and matched directional controls. For $X \to Y$, gain is $\mathrm{CE}(X) - \mathrm{CE}(Y)$, so positive values favor Y.

Comparison	Median gain	Improved
Identity-audited comparisons ($n = 78$)		
$E \to E + R_\Pi$	0.0815	78/78
$I \to E + R_\Pi$	0.0803	76/78
Full-grid comparisons ($n = 81$)		
$E \to I$	-0.0027	35/81
$E - R_\Pi \to E + R_\Pi$	0.3214	81/81
$E + R_{\mathrm{rand}} \to E + R_\Pi$	0.0820	81/81

Table 11: Complete residual-dose sweep over the 81-configuration grid. Gain is cross-entropy at the zero-dose reference minus cross-entropy after the intervention; positive values indicate improvement.

Dose α	Median CE gain	Improved / total
-1.0	-0.19796	0/81
-0.5	-0.07534	0/81
0.0	0	baseline
$+0.5$	+0.02834	76/81
$+1.0$	+0.02241	62/81
$+1.5$	+0.00046	41/81
$+2.0$	-0.04011	24/81

Table 12: Structured-null audit by suite. Pass counts cells whose real span beats the fifth percentile of the matched null.

Suite	n	True q_{50}	Struct. q_{50}	Pass
A	27	0.6281	0.9900	27/27
B	27	0.1095	0.9419	27/27
C	27	0.1782	0.9583	27/27

Table 13: Exact-$C_\perp$ behavioral control at rank 8, clustered by suite and backbone. Entries are medians over three seeds; specificity is the natural minus norm-matched random slope.

Suite	Backbone	$q_\perp$	Endpoint NLL gain	Spec. NLL	Spec. CE	Spec. Acc.	Positive NLL / CE / Acc.
A	Llama-3.1-8B	0.627	+0.11000	+0.14048	+0.13547	+0.04909	3/3, 3/3, 3/3
A	Qwen2.5-1.5B	0.467	+0.07590	+0.10054	+0.09960	+0.03672	3/3, 3/3, 3/3
A	Qwen2.5-7B	0.528	+0.04350	+0.05982	+0.05936	+0.01497	3/3, 3/3, 3/3
B	Llama-3.1-8B	0.100	+0.04557	+0.04479	+0.04142	+0.05212	3/3, 3/3, 3/3
B	Qwen2.5-1.5B	0.083	+0.04136	+0.05157	+0.03244	+0.01152	3/3, 3/3, 3/3
B	Qwen2.5-7B	0.096	+0.03708	+0.03969	+0.03880	+0.00711	3/3, 3/3, 3/3
C	Llama-3.1-8B	0.184	+0.26549	+0.28536	+0.28138	+0.11471	3/3, 3/3, 3/3
C	Qwen2.5-1.5B	0.172	+0.03887	+0.04885	+0.04360	-0.00195	3/3, 3/3, 1/3
C	Qwen2.5-7B	0.148	+0.05845	+0.07000	+0.06998	+0.04987	3/3, 3/3, 3/3

Table 14: Carrier attribution over 18 cells. The final column reports the number of cells in which OPT has larger function-space R^2 than the indicated carrier. OPT exceeds RAW in 17/18 cells but exceeds CROSS in only 11/18, so the result supports a qualified rather than universal functional-attribution claim.

Carrier	Param.-R^2	Func.-R^2	OPT $>$ row
RAW	0.0108	0.1226	17/18
OPT	0.5696	0.2404	–
CROSS	0.0369	0.2248	11/18
FULL	0.5699	0.2425	–

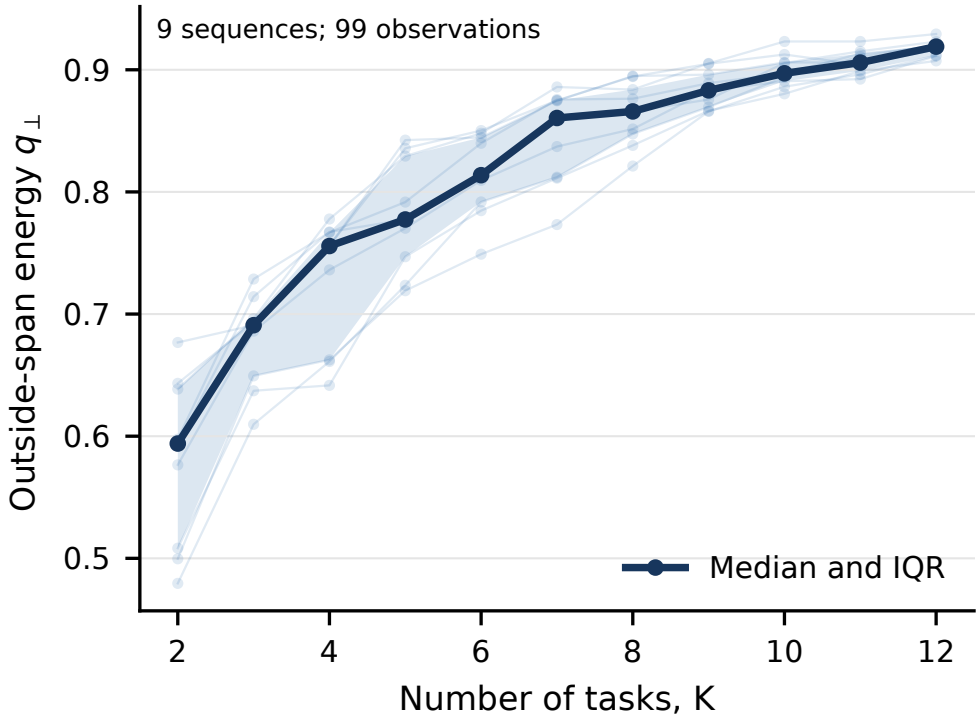


Figure 9: **Outside-span energy across replicated task-count sequences.** The nine light curves show the observed outside-span energy fraction $q_\perp$ for each task-order–seed sequence at $K = 2, \ldots, 12$ (99 observations in total). The dark curve and shaded band show the median and interquartile range across sequences at each task count.

Table 15: Held-out prediction error for larger task compositions. The controlled test predicts $K = 4, 5$ from $K = 2, 3$; the replicated LLM test predicts $K = 10, 11, 12$ from $K = 2, \ldots, 9$ in each of nine task-order–seed sequences. Coefficients are calibrated separately within each controlled configuration and LLM sequence. LLM task counts share tasks and checkpoints. Replicated entries are median per-sequence RMSE [IQR]; lower is better.

Predictor	**Controlled RMSE**	**Fixed sequence**	**9-sequence median [IQR]**
Pairwise accumulation model	**0.03430**	**0.005473**	**0.01170 [0.00547, 0.01365]**
Last-value baseline	0.08762	0.020207	0.02270 [0.01765, 0.03382]
Linear-q baseline	0.06236	0.093310	0.09493 [0.09105, 0.10343]

Table 16: Non-AdamW optimizer control. Each entry is an aggregate over three seeds for the indicated backbone and optimizer.

Backbone	Optimizer	$q_\perp$	ρ_R	$\cos^2$	Natural gain	$-R$ gain	Natural$-$random $q_{.95}$
Qwen2.5-1.5B	Memoryless Adam	0.660	+0.980	0.959	+0.0456	-0.2140	+0.0968
Qwen2.5-1.5B	RMS-only	0.521	+0.985	0.971	+0.0390	-0.1380	+0.0593
Qwen2.5-1.5B	Tensor-normalized	0.571	+0.972	0.946	+0.0885	-0.2326	+0.1420
Qwen2.5-7B	Memoryless Adam	0.773	+0.982	0.975	+0.0362	-0.4106	+0.0396
Qwen2.5-7B	RMS-only	0.648	+0.984	0.978	+0.0299	-0.1267	+0.0261
Qwen2.5-7B	Tensor-normalized	0.746	+0.972	0.964	+0.0504	-0.2807	+0.0815
Llama-3.1-8B	Memoryless Adam	0.880	+0.979	0.968	+0.0655	-0.4797	+0.0982
Llama-3.1-8B	RMS-only	0.796	+0.982	0.966	+0.0383	-0.3454	+0.1195
Llama-3.1-8B	Tensor-normalized	0.844	+0.970	0.948	+0.1021	-0.4629	+0.2973

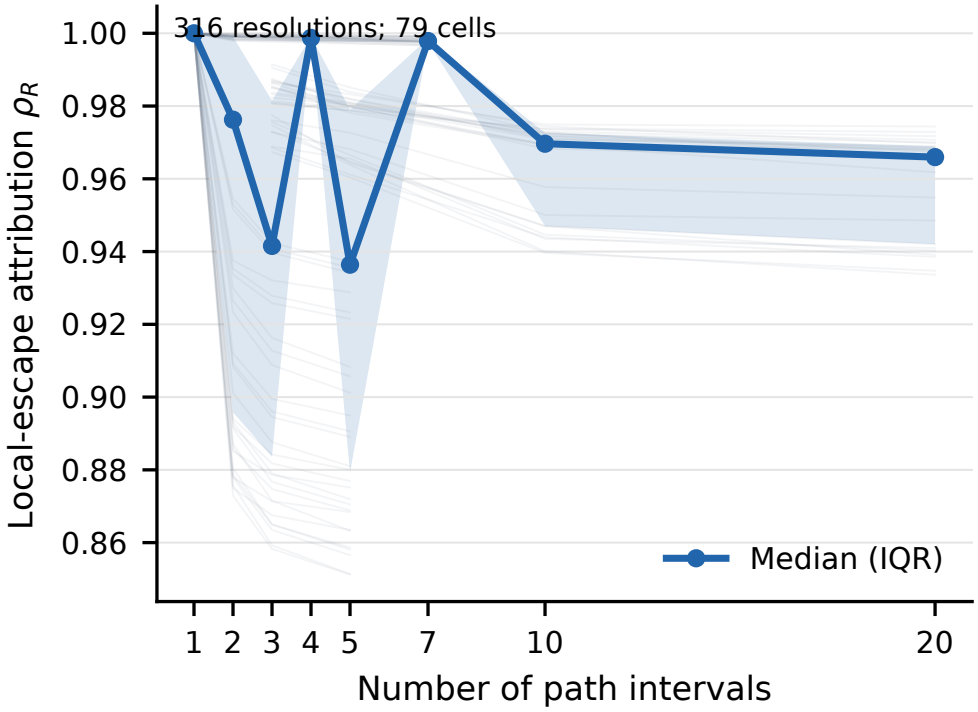


Figure 10: Partition-resolution sensitivity of the optimized path decomposition over all 79 audit-passing cells. Each resolution reuses nested checkpoints; no interpolated checkpoints are introduced.

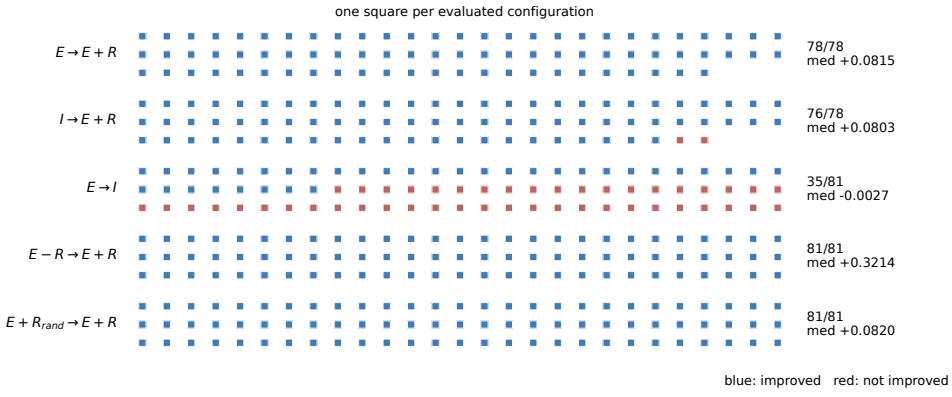


Figure 11: Complete binary outcome audit for functional recovery and directional controls. Each square is one evaluated configuration; the first two rows use 78 identity-audited cells and the remaining rows retain their 81-cell denominators. Exact median gains are printed beside each row.

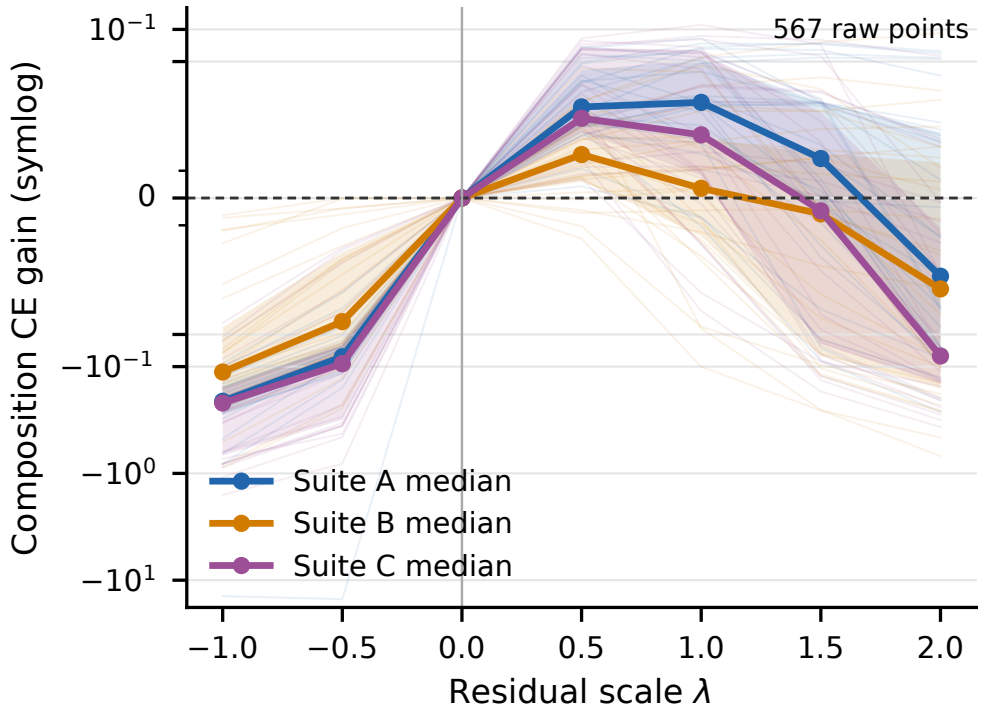


Figure 12: Seven-dose response. The final file contains 81 raw trajectories and suite-level median/interquartile summaries; the fallback is the existing aggregate curve.

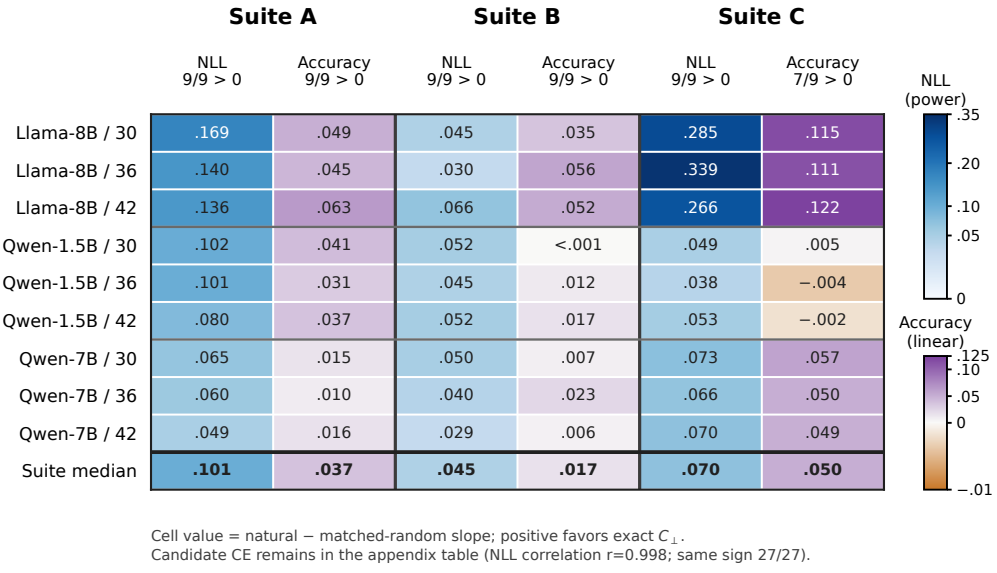
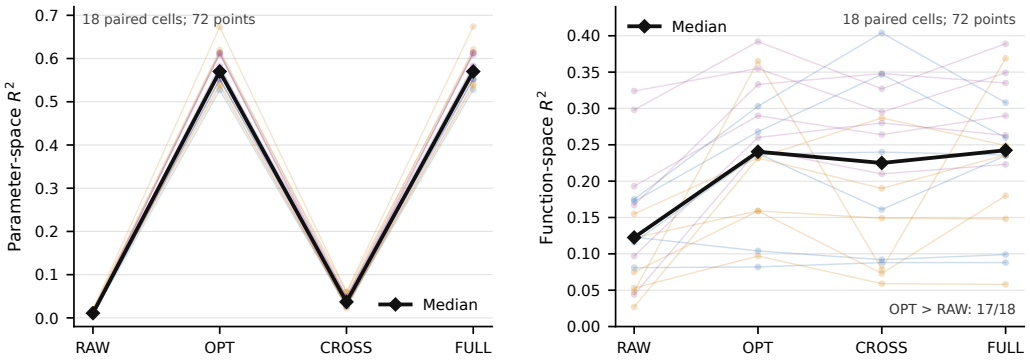


Figure 13: Behavioral specificity of the exact endpoint residual relative to a norm-matched random outside-span direction.



(a) Parameter-space attribution

(b) Function-space attribution

Figure 14: Optimizer-state carrier analyses. The function-space result is boundary evidence rather than a universal carrier-dominance claim.

Table 17: Dose-response robustness by suite and backbone. Gains are medians [IQR] over three ranks and three seeds; positive values indicate lower CE than the zero-dose reference. The last column reports counts whose best tested dose is $\alpha = 0.5/1/1.5/2$, respectively.

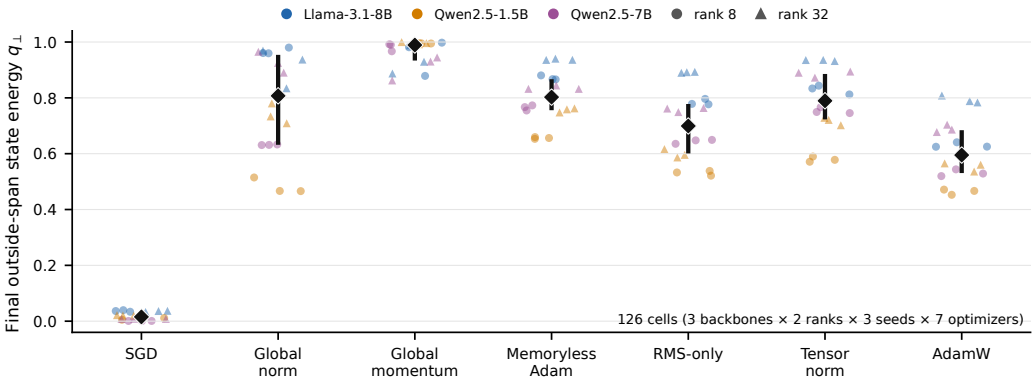
Suite	Backbone	n	Gain at 0.5	Improved	Gain at 1	Improved	Best-dose counts
A	Llama-3.1-8B	9	+0.045 [+0.042, +0.058]	9/9	+0.051 [+0.020, +0.068]	8/9	6/2/1/0
A	Qwen2.5-7B	9	+0.026 [+0.023, +0.033]	9/9	+0.035 [+0.020, +0.035]	7/9	4/4/1/0
A	Qwen2.5-1.5B	9	+0.033 [+0.030, +0.039]	9/9	+0.038 [+0.034, +0.054]	9/9	2/5/2/0
B	Llama-3.1-8B	9	+0.022 [+0.015, +0.035]	8/9	+0.004 [-0.015, +0.022]	6/9	5/3/0/1
B	Qwen2.5-7B	9	+0.016 [+0.006, +0.026]	7/9	+0.010 [-0.004, +0.034]	6/9	3/2/2/2
B	Qwen2.5-1.5B	9	+0.008 [+0.008, +0.019]	7/9	-0.008 [-0.023, +0.012]	4/9	5/2/1/1
C	Llama-3.1-8B	9	+0.066 [+0.065, +0.072]	9/9	+0.063 [+0.062, +0.081]	9/9	6/3/0/0
C	Qwen2.5-7B	9	+0.028 [+0.026, +0.033]	9/9	+0.010 [-0.017, +0.035]	6/9	6/3/0/0
C	Qwen2.5-1.5B	9	+0.022 [+0.011, +0.028]	9/9	+0.010 [+0.003, +0.017]	7/9	6/3/0/0

Table 18: Numerical audit of the optimized path decomposition. The effective audit combines signed closure $|\rho_R + \rho_E - 1|$, telescope checks, and final/local orthogonality checks. Closure statistics below are computed over the 79 passing configurations; the two review cases remain visible but are excluded from claimed summaries.

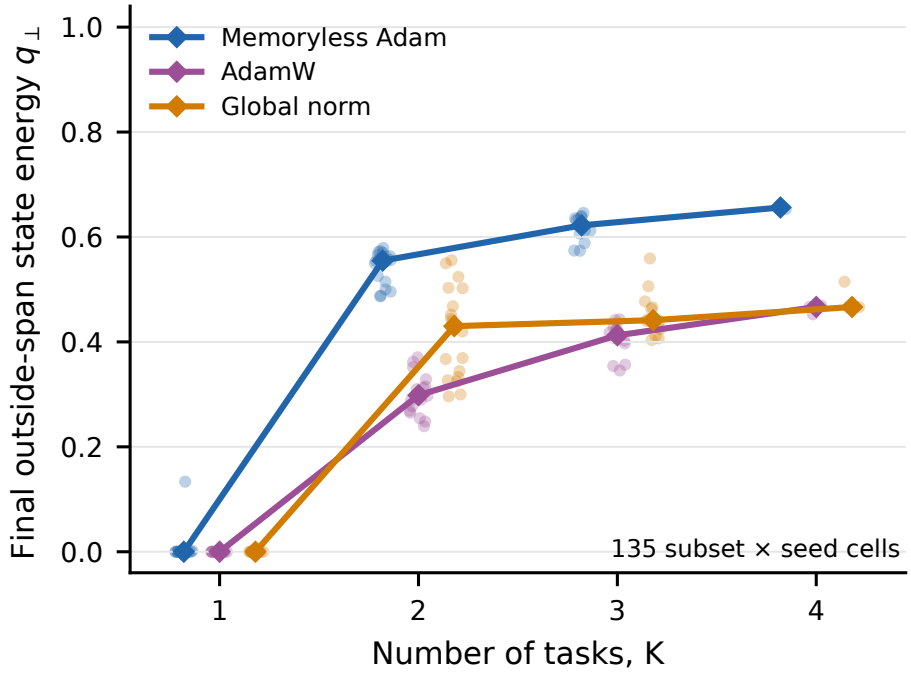
Audit quantity	Result
Collected configurations	81/81
Effective audit PASS	79/81
Numerical REVIEW	2/81
Median signed-closure error (PASS)	1.37×10^{-6}
Maximum signed-closure error (PASS)	4.83×10^{-6}
Review threshold	5×10^{-6}

Table 19: Signed path attribution after the effective numerical audit. Since the decomposition is exact algebraically, $\rho_R + \rho_E = 1$ up to the audited floating-point closure error.

Statistic over 79 PASS cells	ρ_R
Median	0.96596
Interquartile range	[0.90701, 0.99759]
Minimum	0.85124

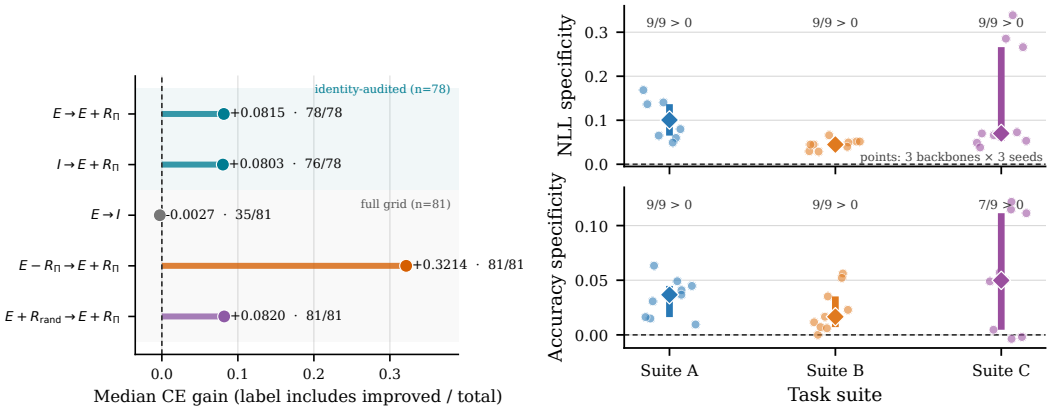


(a) Optimizer-family control



(b) Task-count control

Figure 15: **Optimizer and task-count controls for outside-span energy.** (a) Final outside-span energy across 126 configurations spanning seven optimizers, three backbones, two LoRA ranks, and three seeds. (b) Final outside-span energy across 135 task-subset–seed configurations at $K = 1, \ldots, 4$ on Qwen2.5-1.5B at rank 8. Small points show individual configurations; prominent markers and lines show medians.



(a) Directional-control summary

(b) Behavioral specificity

Figure 16: **Functional recovery depends on the intervention direction.** (a) Each comparison reports its median CE gain and improved-cell count in one aligned forest plot. The top two comparisons use the 78 identity-audited cells; the remaining controls use the full 81-cell grid. (b) Exact-$C_\perp$ minus norm-matched-random specificity across the 27 rank-8 cells for NLL and accuracy, showing raw observations and distribution summaries. Candidate CE is retained in the appendix because it is nearly redundant with NLL.

Table 20: Replicated task-order evaluation of the pairwise accumulation law. Fits are performed separately in each sequence using $K = 2, \ldots, 9$; all three values at $K = 10, 11, 12$ are held out. Errors are median per-sequence RMSE [IQR].

Quantity	Result
Completed task-order–seed sequences	9
Valid task-count observations	99
Median fitted γ [IQR]	0.93484 [0.77200, 1.14969]
Pairwise-law RMSE [IQR]	0.01170 [0.00547, 0.01365]
Last-value RMSE [IQR]	0.02270 [0.01765, 0.03382]
Linear-q RMSE [IQR]	0.09493 [0.09105, 0.10343]
Pairwise law beats last value	8/9 sequences
Pairwise law beats linear-q	9/9 sequences

Table 21: Checkpoint-reused partition sensitivity over the 79 audit-passing cells. Coarsening factors group consecutive primary intervals; interval counts vary because six legacy exact-V1 cells and 73 passing optimized-V2 cells have different checkpoint grids. Differences are paired within cell against its primary partition.

Partition	n	Intervals	Median ρ_R	IQR ρ_R	Median $\lvert\Delta\rho_R\rvert$	Max $\lvert\Delta\rho_R\rvert$
Primary	79	5, 7, or 20	0.96596	[0.90701, 0.99759]	0	0
Coarsened $\times 2$	79	3, 4, or 10	0.96964	[0.91451, 0.99813]	0.00301	0.00827
Coarsened $\times 4$	79	2 or 5	0.97961	[0.92828, 0.99882]	0.01221	0.03020
Coarsened $\times 8$	79	1 or 3	1.00000	[0.98664, 1.00000]	0.02079	0.14876

Table 22: When local escape accumulates along training in the 79 passing configurations. Progress is normalized to $[0, 1]$ within cell; entries are the first piecewise-linear crossing of the stated fraction of final ρ_R.

Attribution reached	Median progress	Q1	Q3
25%	0.02200	0.01562	0.12518
50%	0.25698	0.02940	0.27807
75%	0.40394	0.05101	0.63843

Table 23: Optimizer-family trajectory control over 126 GPU-v2 cells. Every row aggregates 18 configurations (three backbones, two ranks, and three seeds). All entries are medians. "First update" denotes the first valid optimizer update; "all raw/update" denotes the fraction of final outside-span state energy explained by the span of all sampled residual raw gradients/optimizer updates.

Optimizer	n	First raw out	First update out	Final $q_\perp$	Final new-basis frac.	All raw explain	All updates explain
SGD	18	0.000003	0.000003	0.01535	0.00582	0.84222	0.40854
Global norm	18	0.000003	0.000003	0.80723	0.48193	0.04027	0.03897
Global momentum	18	0.000003	0.000003	0.98922	0.56368	0.15810	0.89879
Memoryless Adam	18	0.000003	0.46576	0.80261	0.62166	0.00636	0.05432
RMS-only	18	0.000003	0.32410	0.69924	0.44950	0.01050	0.05421
Tensor norm	18	0.000003	0.02313	0.78943	0.47398	0.00911	0.03917
AdamW	18	0.000003	0.28832	0.59503	0.39836	0.01083	0.56993

Table 24: Task-count trajectory control on Qwen2.5-1.5B, rank 8. Rows aggregate all task subsets and three seeds at a given K; values are medians. The $K = 1$ outside-span quantities are zero by definition.

Optimizer	K	n	First raw out	First update out	Final $q_\perp$	Final new-basis frac.
Memoryless Adam	1	12	0	0	0	0
Memoryless Adam	2	18	0.000004	0.41561	0.55492	0.41946
Memoryless Adam	3	12	0.000095	0.45939	0.62213	0.45709
Memoryless Adam	4	3	0.000004	0.47499	0.65619	0.47755
AdamW	1	12	0	0	0	0
AdamW	2	18	0.000004	0.21995	0.29835	0.18797
AdamW	3	12	0.000095	0.27375	0.41255	0.24337
AdamW	4	3	0.000004	0.28877	0.46664	0.26917
Global norm	1	12	0	0	0	0.0000001
Global norm	2	18	0.000004	0.000004	0.43006	0.31071
Global norm	3	12	0.000095	0.000083	0.44139	0.31790
Global norm	4	3	0.000004	0.000004	0.46656	0.32455