DESA: DELIVERED-ENERGY SPECTRAL ALLOCATION FOR MERGING REINFORCED EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Model merging provides a practical way to combine multiple experts into a single model without retraining or increasing inference cost. However, existing merging methods are largely developed for supervised fine-tuning and do not account for the distinctive structure of reinforcement learning with verifiable rewards (RLVR) updates. We show that RLVR task vectors exhibit two properties that are especially important for merging: *(i)* apparent sparsity under low-precision representation, and *(ii)* substantial heterogeneity in update energy across experts. These properties make sparsification-based heuristics unreliable and cause existing spectral methods to impose an uneven, supply-determined allocation of delivered energy, which strongly affects axis-wise recovery after merging. Motivated by this observation, we propose DESA (**D**elivered-**E**nergy **S**pectral **A**llocation), a spectral merging method that explicitly reallocates expert-level delivered energy while preserving a shared spectral frame. DESA introduces controlled budget equalization across experts together with retention floors that prevent under-allocation to experts whose capabilities are sensitive to reduced budget. Across both language and vision-language RLVR expert pools, DESA consistently outperforms strong baselines in average accuracy and recovery. Our results identify explicit delivered-energy allocation as a key ingredient for merging RL-trained experts.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has rapidly become a standard post-training stage for foundation models, driving gains in domains such as mathematics (Shao et al., 2024), tool use (Qian et al., 2025), embodied decision making (Feng et al., 2025), and other tasks where correctness can be checked automatically. A common practical workflow is to run RLVR separately on different domains to avoid multi-reward conflicts (Bai et al., 2025; Yang et al., 2025a), producing a collection of domain-specialized experts. This creates a natural consolidation problem: how can we combine these reinforced experts into a single model that preserves their complementary strengths? Model merging (Yang et al., 2024b) is an appealing answer because it requires neither original training data nor additional retraining, and it adds no inference-time overhead.

However, most existing model merging methods were designed for supervised finetuning (Yu et al., 2024; Yadav et al., 2023; Zeng et al., 2025), not the update structure induced by RLVR. Sparsification-based methods (Yu et al., 2024; Yadav et al., 2023) assume that useful merge signals live in individual parameters, for example, in large-magnitude entries, sign agreement, or sparsity patterns, while spectral methods (Gargiulo et al., 2025; Marczak et al., 2025) shift to low-rank structure and implicitly determine how merge budget is distributed across experts. However, for RLVR-trained experts, it is unclear whether the coordinate-level signals used by sparsification methods or the default budget split induced by spectral methods are appropriate. We therefore begin by characterizing RLVR task vectors themselves and asking what these properties imply for merging.

This characterization reveals two properties of RLVR task vectors that are especially consequential for merging: *apparent sparsity* under low-precision representation (Zhu et al., 2025) and *substantial heterogeneity* in the overall magnitude of expert updates (see Figure 1). Because RLVR moves weights by only a tiny fraction of the base-model norm, a large fraction of the entries in each task vector quantize to zero under bf16, making magnitude rankings unstable and sign agreement largely uninformative across experts. At the same time, experts can differ substantially in the overall

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

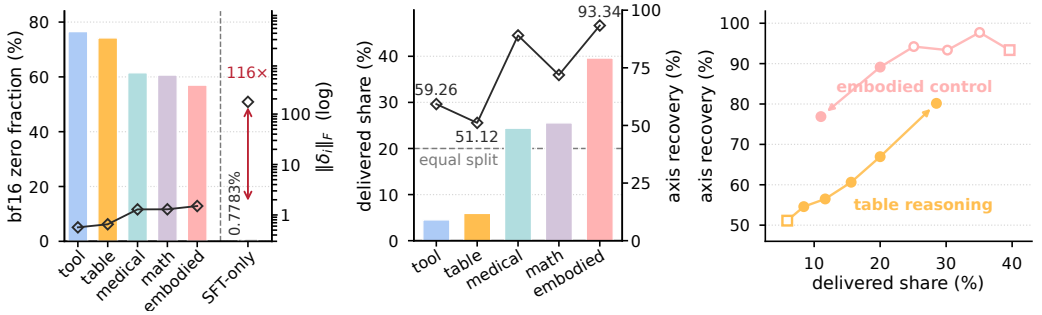

Figure 1: **RLVR task vectors exhibit apparent sparsity and energy heterogeneity.** (a) RLVR task vectors have high bf16 zero fractions because their displacements are extremely small; an SFT-only control moves much farther. (b) Delivered-energy shares are highly uneven and broadly track recovery on each expert axis. (c) Varying delivered share changes axis recovery accordingly, motivating explicit budget allocation; squares mark the TSV operator of (b), filled circles mark where more share bought a clear gain, and hollow circles mark where it bought little.

magnitude of their updates, leading to large differences in total update energy across experts. In spectral merging, the energy retained from each expert therefore determines not only which directions are preserved, but also how much of that expert's update enters the merged model. Existing spectral operators therefore induce an uneven, supply-determined allocation of delivered energy, and we show that this allocation strongly affects axis-wise recovery after merging. This reframes RLVR expert merging as an allocation problem: once updates are represented in spectral form, the key question is not only which directions to keep, but how much delivered energy each expert should receive.

Motivated by this view, we propose DESA (**D**elivered-**E**nergy **S**pectral **A**llocation), a training-free spectral merging method that preserves a shared spectral frame while explicitly reallocating expert-level delivered energy. Unlike prior spectral methods, which inherit a supply-determined budget split from retained spectral mass, DESA formulates merging as a constrained allocation problem over expert-level spectral energy, combining controlled budget equalization with retention floors to avoid overdrawing vulnerable experts. It solves for target expert budgets under a fixed total budget and minimum-retention constraints, then implements them by rescaling the retained singular values in the shared spectral frame. Across both language and vision-language RLVR expert pools, DESA consistently outperforms strong baselines, improving mean recovery by 7.96% on language and 22.57% on vision-language, and in some cases even exceeds joint RL training. Together, these results point to explicit delivered-energy allocation as a key ingredient in merging RL-trained experts. Our contributions are threefold:

- We characterize RLVR task vectors and identify two properties in this regime: apparent sparsity and substantial heterogeneity in the overall magnitude of expert updates.
- We propose DESA, a spectral merging method that explicitly reallocates expert-level delivered energy while preserving a shared spectral frame. It introduces controlled budget equalization together with retention floors to protect vulnerable expert capabilities.
- Across both language and vision-language RLVR expert pools, DESA consistently improves average accuracy and recovery over strong baselines. Our results identify explicit delivered-energy allocation as a key ingredient for merging RL-trained experts.

2 Related Work

Model Merging. Model merging studies how to combine multiple expert checkpoints into a single model that aims to retain their complementary capabilities (Ilharco et al., 2022; Yang et al., 2024b; Wortsman et al., 2022). We consider the standard setting, where all experts are derived from the same initialization and are merged through their task vectors. Existing methods can be broadly organized into three families. *(i) Linear interpolation* adds scaled task vectors back to the base under a single global coefficient (Ilharco et al., 2022; Wortsman et al., 2022). *(ii) Sparsification-based* methods operate coordinate by coordinate, trimming redundant or sign-conflicting entries of each task vector before combination (Yadav et al., 2023; Yu et al., 2024; He et al., 2024b; Yuan et al., 2026; Zeng et al., 2025; Wu et al., 2026). *(iii) Spectral and subspace* methods merge in the SVD bases of the task

vectors, with the retained singular structure implicitly determining how merge budget is distributed across directions (Stoica et al., 2025; Gargiulo et al., 2025; Marczak et al., 2025; Zhu et al., 2026; Li et al., 2026; Zhang et al., 2026). DESA also merges in the spectral domain, but differs in that it explicitly allocates the merging budget according to the energy each expert delivers into the merged model, rather than inheriting that allocation from retained spectral energy.

Post-training with RLVR. Reinforcement learning has become the standard stage of LLM post-training (Yang et al., 2025a; Guo et al., 2025). Early pipelines optimize a reward learned from human preferences to align instruction-following behavior (Ouyang et al., 2022; Rafailov et al., 2023); recent ones replace that reward with programmatic verifiers and use RL with verifiable rewards to sharpen reasoning, mathematics, and code (Shao et al., 2024; Guo et al., 2025; Lambert et al., 2024; Zheng et al., 2025; Yu et al., 2025; Wu et al., 2025). Since each run specializes one shared base model on one verifiable domain, practitioners accumulate exactly the ingredients a merging method consumes.

How RL moves weights, however, differs from supervised finetuning. RL updates appear extremely sparse, concentrating in a small subnetwork and leaving most parameters untouched (Mukherjee et al., 2025; Yuan et al., 2026). Zhu et al. (2025) attribute this sparsity to a geometric bias rather than to genuine locality: KL-anchored RL steps are steered off the principal directions of the pretrained weights into low-curvature, spectrum-preserving subspaces, whereas SFT targets principal weights and distorts the spectrum. RL updates are also low-dimensional in time; the top singular subspace of the update accounts for nearly all of the reasoning gain and evolves close to linearly during training (Cai et al., 2025). Together, these findings indicate that RL post-training leaves a distinctive spectral signature in the weights, which motivates designing a merging algorithm specifically for RLVR post-trained experts.

3 Merging RLVR Experts as Budget Allocation

3.1 Preliminaries and Notation

Task vectors. Expert i's task vector is its displacement from the shared initialization, $\boldsymbol{\delta}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_0$ (Ilharco et al., 2022), where all N experts descend from the same base model $\boldsymbol{\theta}_0$. A merging operator maps $\{\boldsymbol{\delta}_i\}$ to one merged displacement $\boldsymbol{\Delta}_m$, giving the merged model $\boldsymbol{\theta}_0 + \boldsymbol{\Delta}_m$.

Merging signals and delivered budget. Existing merging methods differ mainly in whether they rely on coordinate-level signals, such as magnitude or sign agreement, or instead merge low-rank spectral structure. To quantify how much of each expert enters the merged model across such operators, let $E_i = \|\boldsymbol{\delta}_i\|_F^2$ denote the energy of expert i's task vector, and write the merged update as a sum of expert-attributed terms, $\boldsymbol{\Delta}_m = \sum_i \boldsymbol{c}_i$, where $\boldsymbol{c}_i$ denotes the expert-attributed contribution of expert i to the merged update. We then use the Frobenius energy $\|\boldsymbol{c}_i\|_F^2$ as an operational measure of delivered budget. The corresponding delivered-budget share for expert i is $\mathrm{share}_i = \frac{\|\boldsymbol{c}_i\|_F^2}{\sum_j \|\boldsymbol{c}_j\|_F^2}$. For spectral merging, this share coincides with the retained spectral-energy share of each expert in the pooled decomposition.

Recovery. Every domain is scored against the expert trained for it. We report *recovery* (He et al., 2025), $(\text{merged} - \text{base})/(\text{expert} - \text{base})$, the fraction of that expert's own RLVR gain on its domain that survives merging: 100% matches the expert and 0% matches the base model. Values above 100% are possible when merging exceeds the corresponding expert.

3.2 RLVR Task Vectors: Signals and Energy

In this section, we characterize RLVR task vectors through two properties: apparent sparsity and energy heterogeneity, and study what these properties imply for existing merging operators. We show that coordinate-based rules become unreliable in this regime, while spectral operators implicitly allocate an uneven delivered-energy budget across experts, which in turn affects domain-wise recovery.

Apparent Sparsity. Recent work reports that RLVR task vectors are sparse (Zhu et al., 2025; Yuan et al., 2026; Mukherjee et al., 2025), and Zhu et al. (2025) attribute that sparsity to bf16 quantization. We observe the same phenomenon and refer to it as *apparent sparsity*, since it is driven by representation precision relative to the small magnitude of RLVR displacements. Across experts,

72.48% of task-vector coordinates quantize to zero in bf16 on average. The RLVR displacements are extremely small, averaging 0.1268% of the base model's Frobenius norm across experts, which leaves many coordinates below bf16 resolution. Consistent with this interpretation, the zero fraction decreases over training as the displacement grows (Figure 1a). By contrast, an SFT-only checkpoint trained from the same $\boldsymbol{\theta}_0$ has displacement 27.86% of the base model's Frobenius norm, over two orders of magnitude larger, and shows almost no such quantization loss.

This apparent sparsity makes coordinate-level merging signals unreliable. Magnitude-based selection assumes a meaningful ordering of coordinates, but only 27.52% remain non-zero in bf16, leaving most entries tied at zero. Sign-based arbitration is similarly weak: pairwise sign agreement is only 0.5077 across expert pairs, i.e., essentially random, and task vectors are nearly orthogonal (cosine similarity 0.0018). Thus, low-precision representation undermines magnitude cues, while the geometry of the task vectors undermines sign cues. More importantly, neither provides an explicit notion of how much merge budget each expert should receive. This motivates moving to spectral representations, where merge budget is expressed through energy aggregated over singular directions rather than quantized coordinates.

Heterogeneity and Delivered Energy. RLVR updates are small in absolute scale, but their magnitudes vary substantially across experts: the task vectors in our pool differ in Frobenius norm by a factor of 2.65, hence in energy E_i by 7.02. This heterogeneity matters especially for spectral merging. Spectral operators are usually motivated as preserving, for each expert, the directions that carry most of its update (Gargiulo et al., 2025; Marczak et al., 2025; Zhu et al., 2026). More importantly, in spectral merging, the retained spectral energy of each expert also determines how much of its update enters the merged model. In this sense, retaining energy is also allocating budget. For $\boldsymbol{\Delta}^{\mathrm{TSV}}$, the resulting allocation has a simple closed form. The singular values of $\boldsymbol{\delta}_i$ satisfy $\sum_j s_{ij}^2 = E_i$, so the energy expert i brings through its own top-k directions is $E_i \,\mathrm{ef}_k(i)$, where $\mathrm{ef}_k(i)$ is the fraction of that expert's energy those directions carry, and therefore $\mathrm{share}_i = E_i \,\mathrm{ef}_k(i) \big/ \sum_j E_j \,\mathrm{ef}_k(j)$. For TSV, this delivered-budget share coincides exactly with the retained spectral-energy share of each expert's block in the pooled decomposition; we give the short derivation in Appendix A.2. We term this *supply-determined allocation*. Consequently, the heterogeneity of $\|\boldsymbol{\delta}_i\|_F$ propagates to the budget allocation, modulated only by the degree of concentration of the respective updates.

The resulting allocation is highly uneven: across our pool, delivered shares range from 4.51% for tool use to 39.62% for embodied control, an $8.8\times$ gap. Recovery broadly follows the same ordering (Figure 1b): the least-budgeted experts recover worst, while the largest delivered share corresponds to the strongest recovery. To test this more directly, we vary the delivered share assigned to each expert while keeping the expert pool fixed. Within each expert, delivered share and recovery move together (Figure 1c; Spearman $\rho = 0.915$). At the same time, the marginal return of additional budget is not constant across domains. Extra budget yields large recovery gains when a domain remains far below its own expert, but little gain once recovery is already near saturation. Both regimes are visible in Figure 1c: table reasoning improves across the full range of budget we assign to it, whereas embodied control tolerates a substantial budget reduction before its recovery changes appreciably. This reframes RLVR expert merging as an allocation problem. Once updates are represented spectrally, the key decision is not only which directions to retain, but how much delivered energy each expert should receive. Delivered energy is thus a strong predictor of domain-wise behavior, while the value of that energy is domain-dependent. DESA is designed to address exactly this mismatch.

4 DESA: Delivered-Energy Spectral Allocation

The analysis in Section 3 reframes RLVR expert merging as a budget-allocation problem in spectral space. DESA follows the standard shared spectral construction, but replaces its implicit supply-determined allocation with an explicit expert-level reallocation rule.

4.1 Shared spectral frame

For each weight tensor, let $\boldsymbol{\delta}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_0$ denote expert i's task vector on that tensor. We compute a rank-k truncated SVD:

$$\boldsymbol{\delta}_i \approx \boldsymbol{U}_i \,\mathrm{diag}(\boldsymbol{\sigma}_i)\boldsymbol{V}_i, \tag{1}$$

Algorithm 1 DESA: Delivered-Energy Spectral Allocation

Require: $\boldsymbol{\theta}_0, \{\boldsymbol{\theta}_i\}_{i=1}^N, k, \beta, \{\phi_i\}_{i=1}^N, s_c$
1: $\boldsymbol{\delta}_i \leftarrow \boldsymbol{\theta}_i - \boldsymbol{\theta}_0,\ \boldsymbol{\delta}_i \approx \boldsymbol{U}_i \operatorname{diag}(\boldsymbol{\sigma}_i)\, \boldsymbol{V}_i$ (rank k) $\forall i$
2: Pool $\boldsymbol{U} \leftarrow [\boldsymbol{U}_1 \mid \cdots \mid \boldsymbol{U}_N], \boldsymbol{V} \leftarrow [\boldsymbol{V}_1; \cdots; \boldsymbol{V}_N], \boldsymbol{s} \leftarrow [\boldsymbol{\sigma}_1; \cdots; \boldsymbol{\sigma}_N]$
3: $\boldsymbol{G}_u \leftarrow \boldsymbol{U}^\top \boldsymbol{U}, \boldsymbol{G}_v \leftarrow \boldsymbol{V}\boldsymbol{V}^\top, \widetilde{\boldsymbol{U}} \leftarrow \boldsymbol{U}\boldsymbol{G}_u^{-1/2}, \widetilde{\boldsymbol{V}} \leftarrow \boldsymbol{G}_v^{-1/2}\boldsymbol{V}$
4: $e_i \leftarrow \sum_{a:\mathrm{block}(a)=i} s_a^2\ \forall i$
5: $\alpha_i \leftarrow e_i / \sum_j e_j, u_i \leftarrow e_i^{1-\beta}, \mathrm{floorfrac}_i \leftarrow \phi_i \alpha_i\ \forall i$
6: $\boldsymbol{t} \leftarrow \mathrm{CLAMPRENORM}(\boldsymbol{u}, \mathrm{floorfrac})$ (Algorithm 2)
7: $s_a^{\mathrm{alloc}} \leftarrow \sqrt{t_{\mathrm{block}(a)} / e_{\mathrm{block}(a)}}\, s_a\ \forall a$
8: $\boldsymbol{s}^{\mathrm{alloc}} \leftarrow (\|\boldsymbol{s}\|_2 / \|\boldsymbol{s}^{\mathrm{alloc}}\|_2)\, \boldsymbol{s}^{\mathrm{alloc}}$
9: $\boldsymbol{\Delta}_{\mathrm{DESA}} \leftarrow \widetilde{\boldsymbol{U}} \operatorname{diag}(\boldsymbol{s}^{\mathrm{alloc}})\, \widetilde{\boldsymbol{V}}$
10: **return** $\boldsymbol{\theta}_{\mathrm{merge}} \leftarrow \boldsymbol{\theta}_0 + s_c \cdot \boldsymbol{\Delta}_{\mathrm{DESA}}$

and concatenate the retained factors across experts: $\boldsymbol{U} = [\boldsymbol{U}_1 | \cdots | \boldsymbol{U}_N]$, $\boldsymbol{V} = [\boldsymbol{V}_1; \cdots; \boldsymbol{V}_N]$ and $\boldsymbol{s} = [\boldsymbol{\sigma}_1; \cdots; \boldsymbol{\sigma}_N]$, where $|$ and ; denote horizontal and vertical concatenation, respectively, so that $\boldsymbol{U} \in \mathbb{R}^{m \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r \times n}$ with $r = kN$. After concatenating, we index the retained singular triplets by a global atom index $a \in \{1, \ldots, r\}$. Each atom corresponds to a rank-1 component $s_a \boldsymbol{u}_a \boldsymbol{v}_a^\top$, and $\mathrm{block}(a)$ denotes the expert from which that atom originates.

As in prior spectral merging, the concatenated directions are not mutually orthogonal because each expert is decomposed independently. We therefore form the left and right Gram matrices $\boldsymbol{G}_u = \boldsymbol{U}^\top \boldsymbol{U}, \boldsymbol{G}_v = \boldsymbol{V}\boldsymbol{V}^\top$, and construct shared orthogonal frames via the polar factors $\widetilde{\boldsymbol{U}} = \boldsymbol{U}\boldsymbol{G}_u^{-1/2}, \widetilde{\boldsymbol{V}} = \boldsymbol{G}_v^{-1/2}\boldsymbol{V}$. Here the two normalizations are intentionally asymmetric: $\boldsymbol{U}$ is orthogonalized by right multiplication and $\boldsymbol{V}$ by left multiplication, so that $\widetilde{\boldsymbol{U}}^\top \widetilde{\boldsymbol{U}} = \boldsymbol{I}$ and $\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top = \boldsymbol{I}$. These frames define the common spectral basis in which the merged update is reconstructed. Up to this point, DESA follows the standard TSV-style pooled spectral geometry; our contribution lies in how the retained spectral energy is allocated across experts once this shared frame has been formed.

4.2 Expert-level spectral allocation

In TSV, the retained singular values are used unchanged, so the expert-level budget is exactly the retained spectral energy supplied by each expert. Let $\mathcal{B}_i$ denote the pooled atom indices originating from expert i, and define:

$$e_i := \sum_{a \in \mathcal{B}_i} s_a^2, \qquad \alpha_i := \frac{e_i}{\sum_j e_j}. \tag{2}$$

Collecting these quantities across experts gives the retained-energy vectors $\boldsymbol{e} = (e_1, \ldots, e_N)$ and the corresponding TSV share vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$. As shown in Section 3.2, TSV uses the fixed expert-level allocation $\boldsymbol{t} = \boldsymbol{e}$. For RLVR experts, this supply-determined allocation can be highly uneven and strongly affects domain-wise recovery. DESA keeps the shared spectral frame unchanged and replaces this fixed rule with an explicit expert-level reallocation.

We first define a reference allocation u_i through an equalization exponent $\beta \geq 0$:

$$u_i := e_i^{1-\beta}, \qquad M := \sum_i u_i, \tag{3}$$

where $\boldsymbol{u} = (u_1, \ldots, u_N)$ denote the resulting reference-allocation vector. In implementation, if an expert contributes zero retained energy on a tensor ($e_i = 0$), we treat that block as inactive for reallocation on that tensor and keep its allocated mass at zero; equivalently, the allocation rule is applied only over the active set $\{i : e_i > 0\}$. When $\beta = 0$, the reference allocation is proportional to the supplied energy; when $\beta = 1$, it becomes uniform across experts; and when $\beta > 1$, it favors experts that would otherwise receive less budget.

DESA can further impose retention floors. Let $\phi_i \in [0, 1)$ specify the minimum fraction of each expert's TSV share that must be retained. We then define the target expert-level allocation as:

$$\boldsymbol{t}^\star = \arg\min_{\boldsymbol{t} \in \mathcal{C}} D(\boldsymbol{t} \,\|\, \boldsymbol{u}), \qquad \mathcal{C} = \Big\{ \boldsymbol{t} \in \mathbb{R}^N_{>0} : \sum_i t_i = M,\ \ t_i \geq \phi_i\, \alpha_i\, M \Big\}, \tag{4}$$

where $D(\boldsymbol{t}\|\boldsymbol{u}) = \sum_i \left(t_i \log \frac{t_i}{u_i} - t_i + u_i \right)$ is the generalized Kullback–Leibler divergence. This makes the roles of the two controls explicit: β specifies the reference split we would like, while ϕ_i sets lower bounds on how much of each expert's TSV allocation must be preserved.

The feasible set $\mathcal{C}$ is always nonempty, since the total mass required by the floors is $\sum_i \phi_i \, \alpha_i \, M < \sum_i \alpha_i \, M = M$. Moreover, the solution of Equation (4) has the closed form:

$$t_i^\star = \max\left(\phi_i \, \alpha_i \, M, \; \lambda u_i\right), \tag{5}$$

where $\lambda > 0$ is the unique scalar chosen so that $\sum_i t_i^\star = M$. Appendix A.3 gives the short KKT derivation (Ghojogh et al., 2021). Thus, experts whose reference allocation falls below their floor are clamped to that floor, and the remaining budget is distributed proportionally to $\boldsymbol{u}$ over the unconstrained experts. Algorithm 2 implements this clamp-and-renormalize projection. This also recovers TSV as a special case: when $\beta = 0$ and $\boldsymbol{\phi} = \mathbf{0}$, we have $\boldsymbol{u} = \boldsymbol{e}$ and hence $\boldsymbol{t}^\star = \boldsymbol{e}$.

Finally, DESA converts the expert-level targets back into atom-level coefficients by uniformly rescaling all atoms from the same expert:

$$s_a^{\text{alloc}} = \sqrt{\frac{t^\star_{\text{block}(a)}}{e_{\text{block}(a)}}} \; s_a. \tag{6}$$

Because the rescaling factor depends only on the expert block, DESA changes the expert-level spectral allocation without altering the relative ordering of atoms within that expert, up to the global energy-preserving renormalization described in Section 4.3.

4.3 Energy preservation and final operator

The reallocation in Section 4.2 changes how spectral energy is distributed across experts, but it should not change the total update magnitude. DESA therefore applies a final global renormalization to the atom coefficients $\boldsymbol{s}^{\text{alloc}}$ so that their ℓ_2 norm matches that of the retained singular-value pool, $\|\boldsymbol{s}^{\text{alloc}}\|_2 = \|\boldsymbol{s}\|_2$. This final normalization applies one common global factor to all expert blocks, so it preserves the relative allocation determined by $\boldsymbol{t}^\star$ while restoring the total retained spectral norm. Because $\widetilde{\boldsymbol{U}}$ and $\widetilde{\boldsymbol{V}}$ are orthogonal frames in the sense above, this implies $\|\boldsymbol{\Delta}_{\text{DESA}}\|_F = \|\boldsymbol{s}^{\text{alloc}}\|_2 = \|\boldsymbol{s}\|_2$. Hence β and $\boldsymbol{\phi}$ govern the relative allocation of spectral energy across experts, while a final scalar s_c controls the overall merge strength.

The merged update for the tensor is:

$$\boldsymbol{\Delta}_{\text{DESA}} = \widetilde{\boldsymbol{U}} \, \text{diag}(\boldsymbol{s}^{\text{alloc}}) \widetilde{\boldsymbol{V}}, \qquad \boldsymbol{\theta}_{\text{merge}} = \boldsymbol{\theta}_0 + s_c \cdot \boldsymbol{\Delta}_{\text{DESA}}. \tag{7}$$

Overall, DESA differs from prior spectral merging in one key respect: it keeps the shared pooled spectral frame, but replaces the default supply-determined allocation of spectral energy with an explicit expert-level allocation rule. This is particularly well matched to RLVR experts, whose task vectors are highly heterogeneous in energy and whose downstream recovery is sensitive to how merge budget is distributed across experts.

5 Experiments

5.1 Experimental Setup

Expert Pools and Merging Benchmarks. We evaluate across two independent settings, one language and one vision-language. In each setting, every expert is RLVR-trained from the same shared $\boldsymbol{\theta}_0$. For the language setting, we build upon Qwen2.5-7B-Instruct (Yang et al., 2024a) with five experts, one per domain: tabular reasoning (Yang et al., 2025b), tool use (Qian et al., 2025), embodied agent (Feng et al., 2025), medical (Liu et al., 2025), and mathematics. For the vision-language setting, we build upon OneThinker-SFT (Feng et al., 2026) and train 4 experts in spatial reasoning, video object localization, temporal grounding, and mathematics. Training details of each expert and merging benchmark details for each domain are in the Appendix B.2.

Table 1: **Results on merging five RLVR language experts.** We report task accuracy on each domain and recovery relative to the corresponding oracle expert, where 100% means matching that expert's gain over the shared base model. DESA achieves the best overall performance, improving both average accuracy and mean recovery over prior merging baselines. The low embodied score of the SFT is mainly caused by weak instruction-following, leading to many format-check failures.

Method	Accuracy						Recovery (%)					
	Table	Math	Tool	Emb	Med	Average	Table	Math	Tool	Emb	Med	Average
SFT Base	63.37	42.89	47.46	1.50	60.62	43.17	0.00	0.00	0.00	0.00	0.00	0.00
Oracle	78.18	52.10	61.06	96.67	67.26	71.05	100.00	100.00	100.00	100.00	100.00	100.00
Average	73.28	46.68	50.64	62.83	63.59	59.41	66.92	41.18	23.38	64.45	44.75	48.14
TA	64.55	43.64	39.42	94.83	**68.19**	62.13	7.92	8.19	−59.12	98.07	**114.05**	33.82
TIES	68.88	49.39	51.18	91.83	67.04	65.66	37.16	70.60	27.35	94.92	96.70	65.35
DARE	64.90	43.02	39.48	94.67	67.62	61.94	10.29	1.42	−58.68	97.90	105.45	31.28
DARE-TIES	64.13	41.22	37.17	94.50	68.16	61.04	5.15	−18.09	−75.66	97.72	113.62	24.55
Breadcrumbs	64.22	43.23	39.81	93.67	68.17	61.82	5.69	3.66	−56.25	96.85	113.82	32.75
DELLA	63.92	41.14	35.99	94.33	68.09	60.69	3.67	−18.93	−84.34	97.55	112.55	22.10
Model Stock	64.32	42.79	45.81	3.67	60.73	43.46	6.38	−1.08	−12.13	2.28	1.57	−0.60
MultiSLERP	71.77	46.37	50.36	49.83	63.09	56.29	56.72	37.82	21.32	50.79	37.24	40.78
SCE	69.28	49.69	50.76	91.83	67.57	65.83	39.89	73.87	24.26	94.92	104.74	67.54
TSV	70.94	49.51	55.52	90.33	66.53	66.57	51.12	71.89	59.26	93.34	89.05	72.93
Iso-C	74.08	48.43	56.68	90.83	65.76	67.16	72.32	60.16	67.79	93.87	77.49	74.33
Karcher mean	73.00	45.16	51.86	49.00	63.33	56.47	64.98	24.67	32.35	49.91	40.85	42.55
RAM	70.48	49.38	51.74	91.33	66.64	65.92	47.98	70.50	31.47	94.40	90.75	67.02
RAM+	70.12	49.01	52.09	91.33	66.83	65.88	45.55	66.43	34.04	94.40	93.58	66.80
ISO-Merger	67.41	48.39	48.29	92.83	67.99	64.98	27.23	59.71	6.10	95.97	111.04	60.01
DESA (ours)	**74.44**	**49.83**	**57.06**	**95.00**	66.76	**68.62**	**74.74**	**75.31**	**70.59**	**98.25**	92.54	**82.29**

Compared Baselines. We use the MergeKit (Goddard et al., 2024) to implement all the baselines and compare our method with Iso-C (Marczak et al., 2025), TSV (Gargiulo et al., 2025), RAM (Yuan et al., 2026), SCE (Wan et al., 2025), TIES (Yadav et al., 2023), ISO-Merger (Zhu et al., 2026), Task Arithmetic (Ilharco et al., 2022), DARE (Yu et al., 2024), Breadcrumbs (Davari & Belilovsky, 2024), DARE-TIES (Yu et al., 2024), DELLA (Deep et al., 2024), linear average, Karcher mean (Wang et al., 2026), MultiSLERP, and Model Stock (Jang et al., 2024).

5.2 EXPERIMENTAL RESULTS

Tables 1 and 2 show that DESA consistently achieves the best overall merging performance across both the language and vision-language RLVR expert pools. In the language setting, DESA obtains the highest average accuracy (68.62%) and mean recovery (82.29%), outperforming prior spectral methods such as TSV (Gargiulo et al., 2025) and Iso-Merger (Zhu et al., 2026), as well as sparsification- and linear-based baselines. Its gains are especially pronounced on tabular reasoning and tool use, where recovery improves substantially, demonstrating that explicit expert-level energy reallocation effectively safeguards underserved expert capabilities. In the vision-language setting, DESA again ranks first, reaching 57.34% average accuracy and 114.70% average recovery, outperforming all baselines including joint RL training. We attribute recoveries above 100% to positive cross-expert transfer, where complementary multimodal capabilities reinforce each other. Overall, these results support our central claim: for RLVR experts, merge quality depends not only on retaining spectral directions, but crucially on allocating delivered energy across experts appropriately.

5.3 ABLATION STUDY AND FURTHER ANALYSIS

Retention floors safeguard aggressive equalization. Figure 2 examines DESA's two allocation controls through one-dimensional sweeps. Figures 2a and 2b vary the equalization strength β with and without the retention floor, while Figure 2c sweeps the embodied floor ϕ_{emb} at a fixed $\beta = 1.5$. The results show that β is an effective control knob: moving away from TSV's supply-determined split improves performance over a substantial range, and moderate equalization yields a noticeably higher macro average than TSV (Gargiulo et al., 2025).

Table 2: **Results on merging RLVR vision-language experts.** DESA achieves the best overall performance, obtaining the highest accuracy and recovery among all compared methods.

Method	Accuracy					Recovery (%)				
	Spatial	Video	Temporal	Math	Average	Spatial	Video	Temporal	Math	Average
SFT Base	40.80	37.28	31.94	66.06	44.02	0.00	0.00	0.00	0.00	0.00
Oracle	47.02	59.04	46.73	73.38	56.54	100.00	100.00	100.00	100.00	100.00
Joint GRPO	46.42	57.69	45.31	68.55	54.49	89.18	93.77	90.38	34.09	76.86
Average	43.42	43.46	41.57	68.72	49.29	42.07	28.38	65.13	36.36	42.98
TA	47.03	58.98	43.53	71.88	55.35	100.16	99.72	78.34	79.55	89.44
TIES	44.22	58.81	45.89	69.72	54.66	54.88	98.94	94.32	50.00	74.53
DARE	47.21	58.95	43.42	71.05	55.16	103.00	99.59	77.62	68.18	87.10
DARE-TIES	47.00	58.87	43.25	72.88	55.50	99.63	99.22	76.47	93.18	92.13
Breadcrumbs	46.81	58.84	42.89	70.22	54.69	96.52	99.08	74.01	56.82	81.61
DELLA	46.61	58.91	42.69	70.22	54.61	93.30	99.40	72.71	56.82	80.56
Model Stock	40.68	38.80	33.07	65.72	44.57	−1.93	6.99	7.62	−4.55	2.03
MultiSLERP	44.12	44.22	42.39	68.05	49.69	53.27	31.85	70.68	27.27	45.77
SCE	44.50	59.02	45.91	70.72	55.04	59.49	99.93	94.48	63.64	79.38
TSV	45.73	55.61	46.37	69.72	54.36	79.26	84.23	97.57	50.00	77.76
Iso-C	45.87	54.05	46.35	71.21	54.40	81.46	77.06	97.43	70.45	81.77
Karcher mean	43.88	46.05	43.25	68.72	50.48	49.52	40.31	76.45	36.36	50.66
RAM	44.71	58.72	45.99	71.55	55.24	62.86	98.53	94.97	75.00	82.84
RAM+	44.69	58.62	45.91	71.21	55.11	62.54	98.09	94.48	70.45	81.39
ISO-Merger	45.89	**59.10**	45.52	69.88	55.10	81.73	**100.28**	91.84	52.27	81.53
DESA (ours)	**49.61**	58.32	**46.40**	**75.04**	**57.34**	**141.58**	96.70	**97.77**	**122.73**	**114.70**

At the same time, the benefit of stronger equalization depends on protection against excessive drawdown. Without a floor, performance improves only under moderate equalization and rapidly degrades as β becomes large. With the floor enabled, the macro average varies more smoothly across the full sweep, allowing stronger equalization to remain effective and peak at a larger β. Figure 2b illustrates this stabilizing effect on the embodied domain, whose score drops sharply without protection but remains within a robust, narrow band once floored. Figure 2c confirms this from the floor perspective: increasing the embodied floor lifts performance from the no-floor setting and then saturates once the vulnerable domain is protected. Overall, β controls how aggressively DESA equalizes expert budgets, while the floor ensures that this equalization remains reliably usable.

Uniform allocation improves performance but remains suboptimal. Table 3 compares three operators that share the same TSV spectral frame, differing solely in how retained energy is partitioned among experts. Replacing TSV's supply-determined allocation with a uniform target improves the macro average (from 66.57% to 67.21%), showing that the default TSV split is not optimal for RLVR experts. However, uniform allocation still trails DESA (68.62%). This performance gap indicates that equalizing expert budgets helps but is insufficient: effective allocation must also account for asymmetric sensitivity across experts, which DESA does through stronger equalization with a retention floor.

Table 3: Budget-allocation comparison under a shared spectral frame. Uniform allocation is beneficial, but remains suboptimal relative to DESA.

Method	β	Floor	Average	Δ
TSV	0	✗	66.57	-
Uniform	1	✗	67.21	0.64 ↑
DESA (ours)	1.5	✓	**68.62**	**2.05** ↑

Delivered share reliably predicts domain recovery. Figure 3 relates changes in delivered share to changes in recovered headroom across all merges from the β-sweep and the floor sweep. Each point corresponds to an expert-axis pair, evaluated relative to the TSV reference. The relationship is strongly monotonic within experts, with a Spearman correlation of $\rho = +0.915$: increasing an expert's delivered share tends to recover more headroom, whereas reducing it incurs a performance drop. This empirical evidence firmly supports DESA's core premise that task-specific performance is primarily governed by delivered budget.

Per-expert view under a fixed budget. Figure 4 compares plain TSV ($\beta = 0$, no floor) and DESA ($\beta = 1.5$, with floor) at the same global scale $s_c = 1.0$, which yields an identical total merged-update norm. Under this fixed-budget setting, differences between the two methods reflect reallocation

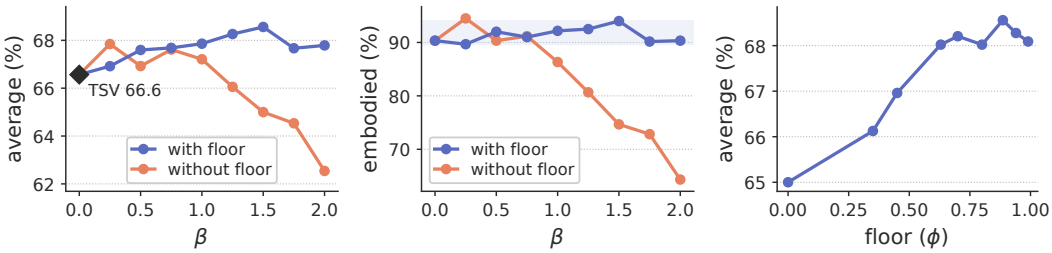

Figure 2: **Effect of the equalization strength and the retention floor.** (a) Macro average versus β. (b) Embodied score versus β. (c) Macro average versus ϕ_{emb} at fixed β. Moderate equalization improves performance, and the retention floor makes stronger equalization more robust.

across experts. DESA effectively shifts budget from experts over-supplied by TSV's default split to under-served ones: tabular reasoning gains 14.8% in delivered share and tool use gains 19.5%, whereas mathematics, medical, and embodied control concede 15.6%, 14.2%, and 4.5%, respectively. This reallocation yields substantive gains: tabular reasoning accuracy rises from 70.9% to 76.5%, tool use improves from 55.5% to 58.4%, and the overall macro average increases from 66.57% to 68.56% under the same total budget.

Embodied control serves as an informative case study. While its delivered share falls by 4.5%, its accuracy still rises from 90.3% to 94.0%, indicating that it is not budget-limited in this range. This observation directly elucidates the design rationale of the retention floor: DESA reallocates budget aggressively where extra energy yields high marginal returns, while the retention floor prevents donor experts from being drawn down past their tolerance thresholds into performance collapse.

Final renormalization is essential for stable merging. Table 4 shows that the final renormalization after the expert-level reallocation in Equation (6) is necessary for stability. Removing it increases the merged displacement from 4.88 to 180.63 in Frobenius norm (37.0×) and collapses the macro average from 68.62 to 4.08. This indicates that reallocating spectral energy across experts can substantially inflate the overall merge magnitude, so a final renormalization is needed to restore a controlled update scale.

Table 4: Ablation of the final renormalization.

Configuration	$\|\Delta\|_F$	Accuracy
w/ renorm	4.88	68.62
w/o renorm	180.63 (37.0×)	4.08

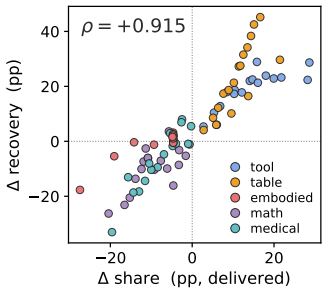

Figure 3: Delivered-share change versus recovered-headroom change. Each point is an expert-axis pair relative to the TSV reference.

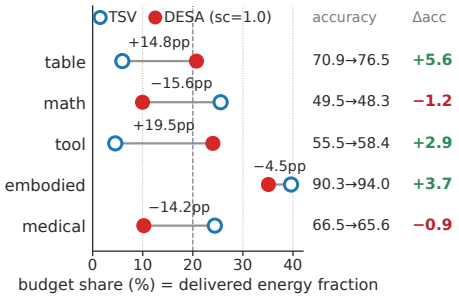

Figure 4: Per-expert view of fixed-budget reallocation. At the same global scale and total merged-update norm, DESA redistributes delivered energy across experts relative to TSV.

6 Conclusion

In this work, we study model merging for RLVR-trained experts through the lens of energy allocation. We show that RLVR task vectors exhibit two key properties: apparent sparsity under low precision and pronounced update energy heterogeneity. Consequently, coordinate-based heuristics break down under quantization-induced sparsity, while spectral methods suffer from their default supply-determined nature, turning energy heterogeneity into severely unbalanced recovery. To resolve this dilemma, we propose DESA, a spectral merging method that explicitly governs delivered energy across experts through controlled budget equalization and capability-protecting retention floors. Across both language and multimodal RLVR pools, DESA consistently outperforms existing baselines, suggesting that explicit budget allocation is a key ingredient for post-RL expert merging.

AI USE STATEMENT

Generative AI tools were used to improve the grammar and readability of the manuscript. We have reviewed all AI-assisted work. The tool was not used to generate research ideas, methodologies, or experimental results. All AI-assisted edits were reviewed and approved by the authors to ensure that they preserved the intended technical meaning. The authors take full responsibility for the final content of this work, including all text edited with the assistance of generative AI.

REFERENCES

Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. Pubhealthtab: A public health table-based dataset for evidence-based fact checking. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 1–16, 2022.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Yuchen Cai, Ding Cao, Xin Xu, Zijun Yao, Yuqing Huang, Zhenyu Tan, Benyi Zhang, Guangzhong Sun, Guiquan Liu, and Junfeng Fang. On predictability of reinforcement learning dynamics for large language models. *arXiv preprint arXiv:2510.00553*, 2025.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*, 2019.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. Hitab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 1094–1110, 2022.

MohammadReza Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision*, pp. 270–287. Springer, 2024.

Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. Della-merging: Reducing interference in model merging through magnitude-based sampling. *arXiv preprint arXiv:2406.11617*, 2024.

Kaituo Feng, Manyuan Zhang, Hongyu Li, Kaixuan Fan, Shuang Chen, Yilei Jiang, Dian Zheng, Peiwen Sun, Yiyuan Zhang, Haoze Sun, et al. Onethinker: All-in-one reasoning model for image and video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5432–5443, 2026.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2025.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5277–5285. IEEE, 2017.

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.

Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. Kkt conditions, first-order and second-order optimization, and distributed optimization: tutorial and survey. *arXiv preprint arXiv:2110.01858*, 2021.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. Arcee's MergeKit: A toolkit for merging large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 477–485, Miami, Florida, US, November 2024. Association for Computational Linguistics.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024a.

Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *arXiv preprint arXiv:2408.13656*, 2024b.

Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang, Tong Zhang, and Han Zhao. Mergebench: A benchmark for merging domain-specialized llms. *Advances in Neural Information Processing Systems*, 38, 2025.

De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. In *European Conference on Computer Vision*, pp. 202–218. Springer, 2024.

Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE transactions on pattern analysis and machine intelligence*, 43(5): 1562–1577, 2019.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Michael Igorevich Ivanitskiy, Rusheb Shah, Alex F Spies, Tilman Räuker, Dan Valentine, Can Rager, Lucia Quirke, Chris Mathwin, Guillaume Corlouer, Cecilia Diniz Behn, et al. A configurable library for generating and manipulating maze datasets. *arXiv preprint arXiv:2309.10498*, 2023.

Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Han. Model stock: All we need is just a few fine-tuned models. In *European Conference on Computer Vision*, pp. 207–223. Springer, 2024.

Sujay Kumar Jauhar, Peter Turney, and Eduard Hovy. Tabmcq: A dataset of general knowledge tables and multiple-choice questions. *arXiv preprint arXiv:1602.03960*, 2016.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*, 2020.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2567–2577, 2019.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 706–715, 2017.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Hongyu Li, Jinyu Chen, Ziyu Wei, Shaofei Huang, Tianrui Hui, Jialin Gao, Xiaoming Wei, and Si Liu. Llava-st: A multimodal large language model for fine-grained spatial-temporal understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8592–8603. IEEE, 2025.

Yayuan Li, Ze Peng, Jian Zhang, Jintao Guo, Yue Duan, and Yinghuan Shi. When shared knowledge hurts: Spectral over-accumulation in model merging. *arXiv preprint arXiv:2602.05536*, 2026.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri 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.

Che Liu, Haozhe Wang, Jiazhen Pan, Zhongwei Wan, Yong Dai, Fangzhen Lin, Wenjia Bai, Daniel Rueckert, and Rossella Arcucci. Beyond distillation: Pushing the limits of medical llm reasoning with minimalist rule-based rl. *arXiv preprint arXiv:2505.17952*, 2025.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6774–6786, 2021.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*, 2022.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D Bagdanov, and Joost Van De Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. *arXiv preprint arXiv:2502.04959*, 2025.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. *Advances in Neural Information Processing Systems*, 38:132119–132138, 2025.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, et al. Fetaqa: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pp. 248–260. PMLR, 2022.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1470–1480, 2015.

Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *Advances in Neural Information Processing Systems*, 38:105523–105553, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. In *International Conference on Learning Representations*, volume 2025, pp. 4501–4519, 2025.

Fanqi Wan, Longguang Zhong, Ziyi Yang, Ruijun Chen, and Xiaojun Quan. Fusechat: Knowledge fusion of chat models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21629–21653, 2025.

Jiayu Wang, Zuojun Ye, and Wenpeng Yin. Functionality-oriented llm merging on the fisher–rao manifold. *arXiv preprint arXiv:2603.04972*, 2026.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. Pmlr, 2022.

Chenrui Wu, Zexi Li, Jiajun Bu, Jiangchuan Liu, and Haishuai Wang. Sparsity curse: Understanding rlvr model parameter space from model merging. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5384–5394, 2026.

Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. *Advances in Neural Information Processing Systems*, 38:143297–143330, 2025.

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, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

E Yang, L Shen, G Guo, X Wang, X Cao, J Zhang, and D Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. arxiv. *arXiv preprint arXiv:2408.07666*, 2024b.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, et al. Mmsi-bench: A benchmark for multi-image spatial intelligence. In *International Conference on Learning Representations*, volume 2026, pp. 157051–157088, 2026.

Zheyuan Yang, Lyuhao Chen, Arman Cohan, and Yilun Zhao. Table-r1: Inference-time scaling for table reasoning. *arXiv preprint arXiv:2505.23621*, 2025b.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR, 21–27 Jul 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale, 2025. *URL https://arxiv. org/abs/2503.14476*, 1(2), 2025.

Xiangchi Yuan, Dachuan Shi, Chunhui Zhang, Zheyuan Liu, Shenglong Yao, Soroush Vosoughi, and Wenke Lee. Behavior knowledge merge in reinforced agentic models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 33007–33028, 2026.

Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. Robustmerge: Parameter-efficient model merging for mllms with direction robustness. *Advances in Neural Information Processing Systems*, 38:71071–71095, 2025.

Han-Chen Zhang, Zi-Hao Zhou, Mao-Lin Luo, Shimin Di, Min-Ling Zhang, and Tong Wei. Dc-merge: Improving model merging with directional consistency. *arXiv preprint arXiv:2603.06242*, 2026.

Jiahui Zhang, Yurui Chen, Yueming Xu, Ze Huang, Jilin Mei, Chunhui Chen, Yanpeng Zhou, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, et al. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *Advances in Neural Information Processing Systems*, 38, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Gspo: Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, 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, Xiaoxia Wu, Yuandong Tian, Shiwei Liu, David Z Pan, et al. Iso: An rlvr-native optimization stack. *arXiv preprint arXiv:2607.19331*, 2026.

A Method Details and Theoretical Properties

A.1 Symbols and Energy Accounting

Section 3.1 of the main text introduces the shared-base merging setting and the basic notation for task vectors, delivered budget, and recovery. To avoid ambiguity in the proofs below, Table 5 summarizes the main symbols used in DESA. We emphasize in particular the distinction between full task-vector energy, retained spectral energy, target allocated energy, and delivered energy after reconstruction.

Table 5: Symbols used in the shared spectral frame and energy accounting. Unless otherwise specified, all quantities are defined at the tensor level.

Symbol	Meaning
$\boldsymbol{\theta}_0$	Shared base model from which all experts are trained.
$\boldsymbol{\theta}_i$	Parameters of expert i.
$\boldsymbol{\delta}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_0$	Task vector of expert i.
N	Number of experts in the pool.
m, n	Shape of the current mergeable weight tensor.
k	Retained rank per expert on the current tensor.
$\boldsymbol{U}_i, \boldsymbol{\sigma}_i, \boldsymbol{V}_i$	Rank-k truncated SVD factors of expert i's tensor-level task vector, $\boldsymbol{\delta}_i \approx \boldsymbol{U}_i \operatorname{diag}(\boldsymbol{\sigma}_i)\boldsymbol{V}_i$.
$\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{s}$	Pooled left factors, right factors, and singular values formed by concatenating the retained factors of all experts.
a	Index of one pooled spectral atom.
$\mathrm{block}(a)$	Expert from which pooled atom a originates.
$\boldsymbol{G}_u = \boldsymbol{U}^\top \boldsymbol{U}$	Left Gram matrix of pooled factors.
$\boldsymbol{G}_v = \boldsymbol{V}\boldsymbol{V}^\top$	Right Gram matrix of pooled factors.
$\widetilde{\boldsymbol{U}} = \boldsymbol{U}\boldsymbol{G}_u^{-1/2}$	Left shared orthogonal frame from the polar factor.
$\widetilde{\boldsymbol{V}} = \boldsymbol{G}_v^{-1/2}\boldsymbol{V}$	Right shared orthogonal frame from the polar factor.
$\boldsymbol{\Delta}_m$	Generic merged tensor update produced by a merging operator on the current tensor.
$\boldsymbol{\Delta}_{\mathrm{DESA}}$	DESA merged tensor update reconstructed in the shared spectral frame.
$E_i = \|\boldsymbol{\delta}_i\|_F^2$	Full task-vector energy of expert i, before spectral truncation.
$e_i = \sum_{a:\mathrm{block}(a)=i} s_a^2$	Retained spectral energy supplied by expert i on the current tensor.
$\alpha_i = e_i / \sum_j e_j$	Expert i's normalized TSV allocation share on the current tensor.
β	Equalization exponent controlling how strongly DESA moves away from the supply-determined split.
$u_i = e_i^{1-\beta}$	Reference allocation used by DESA before imposing retention floors.
$\boldsymbol{\phi}$	Expert-specific retention-floor vector relative to TSV's normalized allocation.
$t_i^\star$	DESA target expert-level allocation after clamp-and-renormalize.
$\boldsymbol{c}_i$	Expert-attributed contribution to the merged tensor update.
$\|\boldsymbol{c}_i\|_F^2$	Delivered energy attributed to expert i after reconstruction.
share_i	Delivered-budget share of expert i, obtained by normalizing $\|\boldsymbol{c}_i\|_F^2$ across experts.
$\boldsymbol{s}^{\mathrm{alloc}}$	Atom coefficients after DESA blockwise rescaling and final renormalization.
s_c	Global scaling coefficient applied after constructing the merged update.

Unless otherwise specified, the quantities in Table 5 are defined at the tensor level. DESA is applied independently to each mergeable tensor, and full-model statistics are obtained by aggregating the corresponding tensor-level quantities. In particular, it is useful to distinguish four levels of accounting throughout the appendix: *(i)* the full expert displacement E_i, which measures how far each expert moves from the shared base; *(ii)* the retained spectral energy e_i, which measures how much of that displacement enters the shared spectral representation; *(iii)* the target allocation $t_i^\star$, which specifies how DESA redistributes retained budget across experts; and *(iv)* the delivered energy $\|\boldsymbol{c}_i\|_F^2$, which measures how much expert-attributed energy is actually present in the reconstructed merged update. Here e_i and $t_i^\star$ are expert-level scalar energies in allocation space, whereas $\boldsymbol{c}_i$ is an attributed update term in parameter space; only its Frobenius energy $\|\boldsymbol{c}_i\|_F^2$ is directly comparable to the allocation variables. We use share_i for the delivered-budget share defined from the attributed reconstruction terms, and α_i for the normalized retained spectral-energy share under TSV; Appendix A.2 shows that these two coincide for TSV.

The proofs in Appendices A.2 to A.5 use this accounting to separate three conceptually distinct choices: the truncation rank k, which determines how much spectral content is retained; the allocation parameters β and $\boldsymbol{\phi}$, which determine how the retained budget is redistributed across experts; and the final global scaling coefficient s_c, which controls the overall merge strength after allocation.

A.2 Delivered Energy in TSV and Its Supply-Determined Allocation

Section 3.2 argues that standard spectral merging does not explicitly choose how much budget each expert should receive; instead, the resulting allocation is inherited from the retained spectral mass of each expert. We now make this statement precise for TSV (Gargiulo et al., 2025) by showing that the delivered energy attributed to expert i is exactly the retained spectral energy of that expert's block in the pooled decomposition.

Proposition 1. *For TSV, let $\mathcal{B}_i$ denote the pooled singular directions contributed by expert i, and define the expert-attributed contribution*

$$\boldsymbol{c}_i = \widetilde{\boldsymbol{U}}_{:,\mathcal{B}_i} \operatorname{diag}(\boldsymbol{s}_{\mathcal{B}_i})\, \widetilde{\boldsymbol{V}}_{\mathcal{B}_i,:},$$

where $\widetilde{\boldsymbol{U}} = \operatorname{Polar}(\boldsymbol{U})$ and $\widetilde{\boldsymbol{V}} = \operatorname{Polar}(\boldsymbol{V})$ are the shared orthogonal frames of Section 4.1. Then the delivered energy of expert i is exactly the retained spectral energy in its block:

$$\|\boldsymbol{c}_i\|_F^2 = \sum_{a \in \mathcal{B}_i} s_a^2.$$

Consequently,

$$\operatorname{share}_i = \frac{\|\boldsymbol{c}_i\|_F^2}{\sum_{i'} \|\boldsymbol{c}_{i'}\|_F^2} = \frac{\sum_{a \in \mathcal{B}_i} s_a^2}{\sum_a s_a^2} = \alpha_i.$$

Proof. By construction, the columns of $\widetilde{\boldsymbol{U}}_{:,\mathcal{B}_i}$ are orthonormal and the rows of $\widetilde{\boldsymbol{V}}_{\mathcal{B}_i,:}$ are orthonormal. Therefore, left and right multiplication preserve the Frobenius norm of the diagonal block:

$$\|\boldsymbol{c}_i\|_F^2 = \left\| \widetilde{\boldsymbol{U}}_{:,\mathcal{B}_i} \operatorname{diag}(\boldsymbol{s}_{\mathcal{B}_i})\, \widetilde{\boldsymbol{V}}_{\mathcal{B}_i,:} \right\|_F^2 = \|\operatorname{diag}(\boldsymbol{s}_{\mathcal{B}_i})\|_F^2 = \sum_{a \in \mathcal{B}_i} s_a^2.$$

Normalizing over experts gives

$$\operatorname{share}_i = \frac{\sum_{a \in \mathcal{B}_i} s_a^2}{\sum_a s_a^2}.$$

Since α_i is defined as

$$\alpha_i = \frac{e_i}{\sum_j e_j}, \qquad e_i = \sum_{a \in \mathcal{B}_i} s_a^2,$$

we obtain $\operatorname{share}_i = \alpha_i$. □

This proposition establishes the precise sense in which TSV is supply-determined: its delivered-budget share is not chosen independently of the retained spectrum, but coincides exactly with each expert's normalized retained spectral energy in the pooled decomposition. In other words, once the rank truncation is fixed, TSV inherits its expert-level budget split directly from the spectral supply of the experts themselves.

A.3 Closed-form Solution of the DESA Allocation Problem

Section 4.2 defines DESA's expert-level allocation as a constrained projection of the reference allocation $\boldsymbol{u}$ onto the feasible set imposed by the retention floor. We now show that this problem is always feasible and that its optimizer has the closed form used in the main text.

Zero-energy blocks. In principle, $e_i = \sum_{a \in B_i} s_a^2$ can be zero if an expert contributes no retained singular mass on a tensor. This case is rare, but we state it here for formal completeness. In that case, DESA treats the corresponding block as inactive and excludes it from the allocation step. For notational simplicity, the derivations below are written over the active set only.

Recall that DESA defines

$$u_i = e_i^{1-\beta}, \qquad M = \sum_i u_i,$$

and solves

$$\boldsymbol{t}^\star = \arg\min_{\boldsymbol{t}\in\mathcal{C}} D(\boldsymbol{t}\|\boldsymbol{u}), \qquad \mathcal{C} = \left\{ \boldsymbol{t} \in \mathbb{R}^N_{>0} : \sum_i t_i = M,\ t_i \geq \phi_i \alpha_i M \right\},$$

where

$$D(\boldsymbol{t}\|\boldsymbol{u}) = \sum_i \left(t_i \log \frac{t_i}{u_i} - t_i + u_i \right).$$

For convenience, define the lower bound

$$\ell_i := \phi_i \alpha_i M.$$

Feasibility. The feasible set $\mathcal{C}$ is always nonempty. Since $\sum_i \alpha_i = 1$ and $\phi_i \in [0, 1)$, the total mass required by the floors is

$$\sum_i \ell_i = \sum_i \phi_i \alpha_i M < M \sum_i \alpha_i = M.$$

Hence the lower bounds consume strictly less than the total mass M, so there always remains positive mass to distribute over the experts.

Proposition 2. *The optimization problem above has a unique solution, and it is given by*

$$t_i^\star = \max(\ell_i, \lambda u_i), \qquad \ell_i = \phi_i \alpha_i M,$$

where $\lambda > 0$ is the unique scalar such that

$$\sum_i t_i^\star = M.$$

Proof. The objective

$$D(\boldsymbol{t}\|\boldsymbol{u}) = \sum_i \left(t_i \log \frac{t_i}{u_i} - t_i + u_i \right)$$

is strictly convex on $\mathbb{R}^N_{>0}$, since each coordinate function

$$t_i \mapsto t_i \log \frac{t_i}{u_i} - t_i + u_i$$

has second derivative $1/t_i > 0$. Because $\mathcal{C}$ is convex and nonempty, the minimizer is unique.

Introduce the lower bounds explicitly and write the problem as

$$\min_{\boldsymbol{t}\in\mathbb{R}^N_{>0}} \sum_i \left(t_i \log \frac{t_i}{u_i} - t_i + u_i \right) \quad \text{subject to} \quad \sum_i t_i = M, \qquad t_i \geq \ell_i\ \forall i.$$

Its Lagrangian is

$$\mathcal{L}(\boldsymbol{t}, \nu, \boldsymbol{\mu}) = \sum_i \left(t_i \log \frac{t_i}{u_i} - t_i + u_i \right) + \nu \left(\sum_i t_i - M \right) + \sum_i \mu_i (\ell_i - t_i),$$

with multipliers $\mu_i \geq 0$. The stationarity condition is

$$\frac{\partial \mathcal{L}}{\partial t_i} = \log \frac{t_i}{u_i} + \nu - \mu_i = 0.$$

If the lower bound is inactive, i.e. $t_i > \ell_i$, complementary slackness gives $\mu_i = 0$, so

$$t_i = u_i e^{-\nu}.$$

Algorithm 2 CLAMPRENORM: projection under retention floor

Require: Utilities $\boldsymbol{u} \in \mathbb{R}^N_{>0}$, floor fractions $\mathrm{floorfrac} \in [0,1)^N$ with $\sum_i \mathrm{floorfrac}_i < 1$
Ensure: Target budgets $\boldsymbol{t} \in \mathbb{R}^N_{\geq 0}$ such that $t_i \geq \mathrm{floorfrac}_i M$ for all i, $\sum_i t_i = M$ where $M = \sum_i u_i$, and the remaining mass is distributed proportionally to $\boldsymbol{u}$ over unconstrained experts

$M \leftarrow \sum_i u_i$
$\boldsymbol{t} \leftarrow \boldsymbol{u}$
$\mathcal{A} \leftarrow \{1, \ldots, N\}$ ▷ active experts
while true **do**
  $\mathcal{V} \leftarrow \{\, i \in \mathcal{A} : t_i < \mathrm{floorfrac}_i M \,\}$
  if $\mathcal{V} = \emptyset$ **then**
    break
  end if
  for each $i \in \mathcal{V}$ **do**
    $t_i \leftarrow \mathrm{floorfrac}_i M$
  end for
  $\mathcal{A} \leftarrow \mathcal{A} \setminus \mathcal{V}$
  $R \leftarrow M - \sum_{i \notin \mathcal{A}} t_i$ ▷ remaining mass
  if $\mathcal{A} = \emptyset$ **then**
    break
  end if
  $Z \leftarrow \sum_{i \in \mathcal{A}} u_i$
  for each $i \in \mathcal{A}$ **do**
    $t_i \leftarrow R \cdot u_i / Z$
  end for
end while
return $\boldsymbol{t}$

Writing $\lambda = e^{-\nu} > 0$, every inactive coordinate satisfies

$$t_i = \lambda u_i.$$

If the lower bound is active, then $t_i = \ell_i$. Therefore every optimal coordinate must have the form

$$t_i^\star = \max(\ell_i, \lambda u_i).$$

It remains to determine λ. Define

$$F(\lambda) = \sum_i \max(\ell_i, \lambda u_i).$$

Each term is continuous and nondecreasing in λ, so F is continuous and nondecreasing. Moreover,

$$F(0) = \sum_i \ell_i < M,$$

while $F(\lambda) \to \infty$ as $\lambda \to \infty$. Hence there exists $\lambda > 0$ such that $F(\lambda) = M$. By strict convexity of the original problem, the optimizer is unique, and so is this value of λ. Therefore

$$t_i^\star = \max(\ell_i, \lambda u_i)$$

is the unique solution. □

This proposition makes the role of DESA's two allocation controls explicit. The exponent β determines the reference split $u_i = e_i^{1-\beta}$, while the floor imposes the lower bound $\ell_i = \phi_i \alpha_i M$. Experts whose reference allocation falls below their floor are clamped to the floor, and the remaining mass is distributed proportionally to $\boldsymbol{u}$ over the unconstrained experts. In particular, DESA recovers TSV as a special case: when $\beta = 0$ and $\boldsymbol{\phi} = \mathbf{0}$, we have $\boldsymbol{u} = \boldsymbol{e}$ and therefore $\boldsymbol{t}^\star = \boldsymbol{e}$.

A.4 CORRECTNESS OF THE CLAMP-AND-RENORMALIZE PROJECTION

Section 4.2 implements the DESA allocation through the CLAMPRENORM procedure in Algorithm 2. We now show that this procedure returns exactly the optimizer characterized in Appendix A.3, rather than a heuristic approximation.

Recall that Algorithm 2 takes as input the reference utilities $\boldsymbol{u} \in \mathbb{R}^N_{>0}$ and the floor fractions $\mathrm{floorfrac} \in [0,1)^N$, where

$$\mathrm{floorfrac}_i = \phi_i \alpha_i, \qquad M = \sum_i u_i,$$

and returns a target allocation $\boldsymbol{t} \in \mathbb{R}^N_{\geq 0}$ such that

$$\sum_i t_i = M, \qquad t_i \geq \mathrm{floorfrac}_i M \quad \forall i.$$

Writing

$$\ell_i := \mathrm{floorfrac}_i M = \phi_i \alpha_i M,$$

the optimizer derived in Appendix A.3 has the form

$$t_i^\star = \max(\ell_i, \lambda u_i)$$

for the unique $\lambda > 0$ satisfying $\sum_i t_i^\star = M$.

Proposition 3. CLAMPRENORM *(Algorithm 2) terminates in at most N iterations and returns the allocation*

$$t_i = \max(\ell_i, \lambda u_i), \qquad \ell_i = \phi_i \alpha_i M,$$

where $\lambda > 0$ is chosen so that $\sum_i t_i = M$. Consequently, it returns the unique optimizer $\boldsymbol{t}^\star$ of Equation (4)*.*

Proof. Let $\mathcal{A}$ denote the active set maintained by Algorithm 2, i.e., the experts that have not yet been clamped to their lower bounds. At initialization, $\mathcal{A} = \{1, \ldots, N\}$ and $\boldsymbol{t} = \boldsymbol{u}$.

At each iteration, the algorithm identifies the set

$$\mathcal{V} = \{\, i \in \mathcal{A} : t_i < \ell_i \,\}$$

of currently active experts whose proportional allocation would violate their lower bound. Every such coordinate is clamped to

$$t_i = \ell_i, \qquad i \in \mathcal{V},$$

and then removed from the active set. The remaining mass is

$$R = M - \sum_{i \notin \mathcal{A}} t_i,$$

and this mass is redistributed over the updated active set in proportion to $\boldsymbol{u}$, namely

$$t_i = R \, \frac{u_i}{\sum_{j \in \mathcal{A}} u_j}, \qquad i \in \mathcal{A}.$$

Thus, after every redistribution step, all unclamped coordinates satisfy

$$t_i = \lambda u_i \qquad \text{for all } i \in \mathcal{A},$$

for one common scalar

$$\lambda = \frac{R}{\sum_{j \in \mathcal{A}} u_j},$$

while all coordinates outside $\mathcal{A}$ satisfy

$$t_i = \ell_i.$$

Therefore, throughout the procedure, the iterate $\boldsymbol{t}$ always has the structure

$$t_i = \begin{cases} \ell_i, & i \notin \mathcal{A}, \\ \lambda u_i, & i \in \mathcal{A}, \end{cases}$$

for some active set $\mathcal{A}$ and scalar $\lambda > 0$. If some active coordinate still violates its lower bound, it is added to $\mathcal{V}$ and clamped in the next pass. Since at least one index is removed from $\mathcal{A}$ whenever $\mathcal{V} \neq \varnothing$, the active set can only shrink, and the algorithm terminates in at most N iterations.

At termination, there is no active coordinate with $t_i < \ell_i$. Hence the returned vector satisfies

$$t_i \geq \ell_i \quad \forall i, \qquad \sum_i t_i = M,$$

and has the form

$$t_i = \max(\ell_i, \lambda u_i)$$

for some $\lambda > 0$. By Proposition 2, this is exactly the unique optimizer of Equation (4). Therefore Algorithm 2 returns $\boldsymbol{t}^\star$. □

The proposition shows that CLAMPRENORM is not an approximation to the DESA allocation problem: it computes the exact generalized-KL projection onto the simplex with retention floors. Operationally, the procedure repeatedly clamps experts whose reference allocation falls below their floor and redistributes the remaining mass proportionally to the reference utilities over the unconstrained experts.

A.5 ENERGY PRESERVATION AND THE ROLE OF FINAL RENORMALIZATION

The expert-level allocation in DESA is first realized through the blockwise rescaling

$$s_a^{\text{alloc}} = \sqrt{\frac{t^\star_{\text{block}(a)}}{e_{\text{block}(a)}}}\, s_a,$$

followed by the final global renormalization

$$\boldsymbol{s}^{\text{alloc}} \leftarrow \gamma \boldsymbol{s}^{\text{alloc}}, \qquad \gamma := \frac{\|\boldsymbol{s}\|_2}{\|\boldsymbol{s}^{\text{alloc}}\|_2}.$$

The following proposition formalizes the role of this final step.

Proposition 4. *Let $\boldsymbol{s}^{\text{alloc}}$ denote the atom coefficients after blockwise rescaling and before the final global renormalization. Then:*

1. *for each expert i, the pre-renormalization block energy equals the DESA target,*

$$\sum_{a \in \mathcal{B}_i} (s_a^{\text{alloc}})^2 = t_i^\star;$$

2. *after the renormalization $\boldsymbol{s}^{\text{alloc}} \leftarrow \gamma \boldsymbol{s}^{\text{alloc}}$, the block energies remain proportional to $\boldsymbol{t}^\star$, i.e.*

$$\sum_{a \in \mathcal{B}_i} (\gamma s_a^{\text{alloc}})^2 = \gamma^2 t_i^\star;$$

3. *the total retained spectral norm is restored,*

$$\|\gamma \boldsymbol{s}^{\text{alloc}}\|_2 = \|\boldsymbol{s}\|_2;$$

4. *the reconstructed merged update satisfies*

$$\|\boldsymbol{\Delta}_{\text{DESA}}\|_F = \|\boldsymbol{s}\|_2.$$

Proof. For any expert i, the blockwise rescaling factor depends only on the expert block, so

$$\sum_{a \in \mathcal{B}_i} (s_a^{\text{alloc}})^2 = \sum_{a \in \mathcal{B}_i} \frac{t_i^\star}{e_i} s_a^2 = \frac{t_i^\star}{e_i} \sum_{a \in \mathcal{B}_i} s_a^2 = t_i^\star,$$

since $e_i = \sum_{a \in \mathcal{B}_i} s_a^2$. This proves the first claim.

After the final renormalization, every atom coefficient is multiplied by the same scalar γ, so

$$\sum_{a\in\mathcal{B}_i}(\gamma s_a^{\mathrm{alloc}})^2 = \gamma^2 \sum_{a\in\mathcal{B}_i}(s_a^{\mathrm{alloc}})^2 = \gamma^2 t_i^\star,$$

which proves the second claim. In particular, for any i, j,

$$\frac{\sum_{a\in\mathcal{B}_i}(\gamma s_a^{\mathrm{alloc}})^2}{\sum_{b\in\mathcal{B}_j}(\gamma s_b^{\mathrm{alloc}})^2} = \frac{t_i^\star}{t_j^\star},$$

so the renormalization preserves the relative allocation across experts.

The third claim holds by construction of

$$\gamma = \frac{\|\boldsymbol{s}\|_2}{\|\boldsymbol{s}^{\mathrm{alloc}}\|_2}.$$

Finally, since $\widetilde{\boldsymbol{U}}^\top\widetilde{\boldsymbol{U}} = \boldsymbol{I}$ and $\widetilde{\boldsymbol{V}}\widetilde{\boldsymbol{V}}^\top = \boldsymbol{I}$, left and right multiplication by the shared orthogonal frames preserves the Frobenius norm, and therefore

$$\|\boldsymbol{\Delta}_{\mathrm{DESA}}\|_F = \left\|\widetilde{\boldsymbol{U}}\,\mathrm{diag}(\gamma\boldsymbol{s}^{\mathrm{alloc}})\,\widetilde{\boldsymbol{V}}\right\|_F = \|\mathrm{diag}(\gamma\boldsymbol{s}^{\mathrm{alloc}})\|_F = \|\gamma\boldsymbol{s}^{\mathrm{alloc}}\|_2 = \|\boldsymbol{s}\|_2.$$

This proves the fourth claim. □

Therefore, the final renormalization does not alter DESA's expert-level allocation pattern; it only restores the total retained spectral norm after reallocation. This is why the allocation parameters β and $\boldsymbol{\phi}$ control relative budget across experts, while the scalar s_c remains the sole parameter controlling the overall merge strength. As shown empirically in Table 4, omitting this step can cause severe inflation of the merged-update norm and destabilize performance.

B Additional Experimental Details

B.1 Expert Pools and Training Details

Both expert pools are RLVR-trained from a single shared initialization $\boldsymbol{\theta}_0$, so that every task vector $\boldsymbol{\delta}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_0$ is measured against the same base (Section 3).

Language pool. The language pool starts from Qwen2.5-7B-Instruct (Yang et al., 2024a) and comprises five experts, one per domain. Four of them are the public RLVR checkpoints released by their source works; we train the mathematics expert ourselves, since no matching checkpoint from the same base was available. Table 6 lists the provenance of each expert. For the released checkpoints, we refer the reader to the corresponding papers for full training recipes. Below, we detail only the mathematics expert we train.

Table 6: Language expert pool. All experts share the Qwen2.5-7B-Instruct base. Four are released RLVR checkpoints used as-is; the mathematics expert is trained by us.

Domain	Checkpoint	RL objective	Provenance
Tool use	ToolRL-grpo-cold	GRPO, rule-based reward	released (Qian et al., 2025)
Tabular reasoning	Table-R1-Zero-7B	GRPO, rule-based reward	released (Yang et al., 2025b)
Embodied control	GiGPO-ALFWorld	GiGPO, task reward	released (Feng et al., 2025)
Medical	AlphaMed-7B-instruct-rl	GRPO, rule-based reward	released (Liu et al., 2025)
Mathematics	(this work)	GRPO on DAPO-Math-17k	trained by us

Mathematics expert (trained by us). We train the mathematics expert with GRPO (Shao et al., 2024) on DAPO-Math-17k (Yu et al., 2025). The reward is a rule-based verifier that checks the boxed final answer, with a small format term (weight 0.1) and an overlong-response soft penalty (buffer 4096 tokens, penalty factor 1.0). We disable the KL penalty, apply DAPO Clip-Higher ($\varepsilon_{\mathrm{low}} = 0.2$, $\varepsilon_{\mathrm{high}} = 0.28$), token-mean loss aggregation, and online dynamic sampling that keeps only prompts

whose batch accuracy lies in $(0.01, 0.99)$. Optimization uses AdamW in bf16 with learning rate 1×10^{-6}, weight decay 0.1, and 10 warmup steps; both the parameters and the released weights are kept in bf16. Each step samples 16 responses per prompt at temperature 1.0 and top-p 1.0, with maximum prompt length 2048 and maximum response length 20480. The batch geometry follows DAPO's 512 prompts $\times$ 16 responses on 64 GPUs and is scaled down only by the available GPU count, holding the per-device global batch and the minibatch-to-rollout ratio fixed.

Vision-language pool. The vision-language pool starts from OneThinker-SFT (a Qwen3-VL 8B SFT model) (Feng et al., 2026) and comprises four experts, all trained by us with GRPO (Shao et al., 2024): spatial reasoning, video object localization, temporal grounding, and multimodal mathematics. All four share the same optimizer and rollout settings: AdamW in bf16, learning rate 2×10^{-6}, rollout batch and global batch of 256, tensor-parallel size 1, and the same $(0.01, 0.99)$ dynamic-sampling filter and differ only in their rule-based reward and training data (bounding-box, temporal-interval, or answer-matching signals for the localization, grounding, and reasoning axes respectively). The grounding and mathematics experts are trained for three epochs. The training configuration of joint GRPO is the same as each expert.

Checkpoint selection. We do not perform in-training validation; checkpoints are saved at a fixed cadence, and the released expert is taken at a fixed training step rather than by best-dev selection.

B.2 Evaluation Benchmarks and Metrics

Benchmarks per domain. In the language setting we evaluate tabular reasoning on WTQ (Pasupat & Liang, 2015), HiTab (Cheng et al., 2022), TabMWP (Lu et al., 2022), TabMCQ (Jauhar et al., 2016), FeTaQA (Nan et al., 2022), TabFact (Chen et al., 2019) and PubHealthTab (Akhtar et al., 2022); mathematics on AIME, AMC, MATH-500 (Lightman et al., 2023) and OlympiadBench (He et al., 2024a); tool use on BFCL v3 (Patil et al., 2025); embodied on ALFWorld (Shridhar et al., 2020); and medical on MMLU-Pro-Health (Wang et al., 2024), MedMCQA (Pal et al., 2022), MedQA (Jin et al., 2020), and PubMedQA (Jin et al., 2019). In the vision-language setting we evaluate spatial reasoning on Maze (Ivanitskiy et al., 2023), SPAR-Bench (Zhang et al., 2025) and MMSI-Bench (Yang et al., 2026); VidObjLoc on STVG (Li et al., 2025) and GOT-10k (Huang et al., 2019); Temporal Grounding on ActivityNet (Krishna et al., 2017), Charades (Gao et al., 2017), and ANet-RTL (Huang et al., 2024); and multimodal math on Geometry3K (Lu et al., 2021). Every benchmark is run under one protocol shared by the base model, the experts and all merges. Table 7 lists the benchmarks that constitute each domain in both pools.

Table 7: Benchmarks composing each evaluation domain. Language experts share the Qwen2.5-7B-Instruct base; vision-language experts share the OneThinker-SFT base.

Pool	Domains	Benchmarks
Language	Tabular reasoning	WTQ, HiTab, TabMWP, TabMCQ, FeTaQA, TabFact, PubHealthTab
	Mathematics	AIME, AMC, MATH-500, OlympiadBench
	Tool use	BFCL v3
	Embodied control	ALFWorld
	Medical	MMLU-Pro-Health, MedMCQA, MedQA, PubMedQA
Vision-language	Spatial reasoning	Maze, SPAR-Bench, MMSI-Bench
	Video object localization	STVG, GOT-10k
	Temporal grounding	ActivityNet, Charades, ANet-RTL
	Multimodal mathematics	Geometry3K

Accuracy and Aggregation. Each benchmark is scored by task-specific rule-based correctness (boxed-answer matching for mathematics, label or exact-answer matching for the QA suites, and the benchmark's own verifier for tool use, embodied control, and localization/grounding), reported as a percentage. A domain score is the macro-average of its constituent benchmarks, and the overall score is the unweighted mean over axes: for the five language axes and the analogous macro over the four vision-language axes. We use macro-averaging throughout so that a domain with many benchmarks does not dominate the aggregate.

Recovery. For a domain i with shared base score S_i^{base}, single-expert score S_i^{expert}, and merged-model score S_i^{merged}, recovery is the per-domain fraction of the expert's gain over the base that the merged model retains:

$$\text{recovery}_i = \frac{S_i^{\text{merged}} - S_i^{\text{base}}}{S_i^{\text{expert}} - S_i^{\text{base}}} \times 100\%, \tag{8}$$

where the expert is the domain's own single expert and the base is the shared initialization $\boldsymbol{\theta}_0$. A value of 100% means the merged model matches the standalone expert on that domain. Values above 100% occur when the merged model exceeds the standalone expert (its gain over the base is larger than the expert's own), and values below 0% when merging moves the domain below the base. Because the denominator is the expert's own gain, the ratio is most informative on axes where that gain is substantial and is correspondingly sensitive on axes whose expert barely moves the base.

B.3 Measuring bf16 Apparent Sparsity

For expert i and each weight tensor $\boldsymbol{W}$, we compute the task vector in the operational precision used during model merging: checkpoints stored in `bf16` are cast to and subtracted in `float64`, i.e., $\boldsymbol{\delta}_{i,\boldsymbol{W}} = \boldsymbol{W}_i - \boldsymbol{W}_0$. An entry is strictly counted as zero only if it vanishes identically after subtraction, with no numerical threshold applied. The aggregate zero fraction z_i of expert i over a tensor collection $\mathcal{T}$ is given by:

$$z_i = \frac{\sum_{\boldsymbol{W} \in \mathcal{T}} \#\{\, j : [\boldsymbol{\delta}_{i,\boldsymbol{W}}]_j = 0 \,\}}{\sum_{\boldsymbol{W} \in \mathcal{T}} \text{numel}(\boldsymbol{W})}, \tag{9}$$

where $\mathcal{T}$ denotes the set of all weight tensors across all layers of the model.

B.4 Merging Implementation Details

All methods are applied in the shared-base setting of Section 3: each expert checkpoint is converted to a task vector $\boldsymbol{\delta}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_0$, and the merged model is obtained by adding the merged update back to the same base model $\boldsymbol{\theta}_0$. We perform full-parameter merging, i.e., all parameter tensors with matching name and shape across the base model and all expert checkpoints are included.

For spectral methods, each parameter tensor is processed independently in matrix form. Following TSV (Gargiulo et al., 2025), for a tensor of shape $\text{rows} \times \text{cols}$, we retain $k = \left\lfloor \frac{\min(\text{rows},\text{cols})}{N} \right\rfloor$ singular triplets per expert, where N is the number of experts being merged. DESA and TSV therefore use the same tensor set and the same rank budget; they differ only in how the retained spectral energy is allocated across experts after the shared spectral frame is formed.

For the language expert pool, the final DESA configuration uses $\beta = 1.5$ and $s_c = 1.25$ with a nonzero retention floor applied only to the embodied expert ($\phi_{emb} = 0.88$; all other experts use zero floor). All compared methods are run through MergeKit (Goddard et al., 2024) under the same shared-base and full-parameter setting.

B.5 Baseline Details

We describe each merging baseline and the implementation we use for all comparisons.

TSV (Gargiulo et al., 2025). Task Singular Vectors is our closest spectral predecessor. For each 2D weight of shape $m \times n$ ($r = \min(m, n)$) it takes every expert's top-$k = \lfloor r/N \rfloor$ singular triplets, stacks them into pooled factors $\boldsymbol{U}, \boldsymbol{s}, \boldsymbol{V}$, and reconstructs $\boldsymbol{\Delta} = \text{Polar}(\boldsymbol{U})\ \text{diag}(\boldsymbol{s})\ \text{Polar}(\boldsymbol{V})$, where the polar factors whiten the non-orthogonal pooled frames. Setting $k = \lfloor r/N \rfloor$ lets the N experts jointly fill the retained subspace without exceeding full rank, so k is fixed per tensor rather than tuned. Its limitation, which motivates DESA, is that retained singular values are used unchanged, fixing each expert's delivered budget to its own supply-determined spectral energy.

Iso-C (Marczak et al., 2025). Isotropic merging in the common subspace first sums the experts' task vectors into $\boldsymbol{\Delta}_{\text{TA}}$, takes its SVD $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, replaces the entire spectrum with its mean $\bar{\sigma} = \frac{1}{r}\sum_i \sigma_i$, and reconstructs $\boldsymbol{\Delta} = \bar{\sigma}\, \boldsymbol{U}\boldsymbol{V}^\top$. Flattening every singular value to the same $\bar{\sigma}$ makes the merged

update isotropic in the common subspace, equalising directions rather than experts. The algorithm contains no per-expert coefficient anywhere, so it cannot target how much budget an individual expert delivers, the axis DESA acts on, even though it is among the stronger spectral baselines.

ISO-Merger (Zhu et al., 2026). ISO-Merger inherits the base spectrum and optimises only the singular frames. For each 2D weight it keeps the base singular values $\boldsymbol{S}_0$ fixed, sign-aligns every expert's $\boldsymbol{U}_i, \boldsymbol{V}_i$ to the base, projects the frame displacements onto the Stiefel tangent spaces, masks the trailing modes, solves a ridge-regularised unit-retention system for per-expert coefficients (clipped to $[0, 1.5]$), aggregates them, and retracts via polar factors to form $\boldsymbol{W}^\star = \boldsymbol{U}^\star \boldsymbol{S}_0 \boldsymbol{V}^{\star\top}$; 1D parameters are averaged. By construction the merged model reuses the base spectrum, so it edits directions but not spectral magnitudes and therefore cannot reallocate delivered spectral energy across experts the way DESA does.

Task Arithmetic (Ilharco et al., 2022). Task Arithmetic adds the unweighted sum of task vectors back to the base, $\boldsymbol{\theta}_0 + \lambda \sum_i \boldsymbol{\delta}_i$. In mergekit it is the generalized task-arithmetic operator with no sign consensus and no sparsification, so every coordinate of every expert is kept at full magnitude. This makes it the simplest data-free, training-free merge and the reference point against which interference-reduction methods are measured. Because RLVR task vectors are nearly orthogonal and heterogeneous in norm, the raw sum lets the largest-norm experts dominate, which manifests as the strongly negative tool-use recovery we observe.

TIES (Yadav et al., 2023). TIES reduces interference in three coordinate-level steps: trim each task vector to its top-magnitude entries, elect a per-coordinate majority sign from the sum of signed values, and average only the entries agreeing with the elected sign, normalising by the count of agreeing experts. In mergekit it is the generalized operator with `magnitude` sparsification and `sum` consensus. The design assumes useful signal lives in large-magnitude, sign-consistent coordinates. Under RLVR task vectors—where most coordinates quantise to zero in bf16 and pairwise sign agreement sits near chance—both the magnitude ranking and the sign vote become weak signals.

DARE (Yu et al., 2024). DARE drops task-vector entries with probability $1 - d$ via a Bernoulli mask and rescales survivors by $1/d$ so the update's expectation is preserved, then sums them onto the base with no sign consensus (`random` sparsification, rescale on). The stochastic pruning is meant to thin dense supervised updates while keeping their expected direction. On RLVR experts the update is already apparently sparse, so random dropping mostly discards genuine signal, and the $1/d$ rescaling amplifies the surviving heterogeneity—driving the large negative tool-use recovery we report.

DARE-TIES (Yu et al., 2024). DARE-TIES combines DARE's stochastic drop-and-rescale with TIES's sign election: each task vector is randomly pruned and rescaled by $1/d$, then a majority sign is computed from the summed values and only sign-agreeing survivors are merged (`random` sparsification, `sum` consensus, rescale on). It aims to pair DARE's update thinning with TIES's conflict resolution. Because RLVR task vectors are near-orthogonal with chance-level sign agreement, the consensus step arbitrates essentially random signs on top of already-sparse updates, giving one of the weakest recoveries among the coordinate methods on the language pool.

Breadcrumbs (Davari & Belilovsky, 2024). Model Breadcrumbs keeps a middle band of each task vector: it removes both the largest-magnitude outliers and the smallest entries, retaining a `density` fraction in between, then sums the masked deltas onto the base (`magnitude_outliers` sparsification). Discarding outliers is meant to suppress noisy, disruptive weights while keeping the bulk of the useful update. As with the other magnitude-based masks, this presumes a reliable magnitude ordering; under bf16-quantised RLVR deltas most coordinates are exactly tied at zero, so the retained band is a weak proxy for task-relevant signal.

DELLA (Deep et al., 2024). DELLA replaces hard thresholding with rank-based stochastic pruning: within each row it ranks coordinates by magnitude and assigns Bernoulli keep-probabilities increasing linearly with rank over $[\,d - \epsilon,\, d + \epsilon\,]$, rescales survivors, then applies TIES-style sign consensus and normalisation (`della_magprune`, `sum` consensus). Giving larger weights a higher survival chance is meant to keep important entries more often than uniform DARE. The rank signal it relies

on is exactly what bf16-quantised RLVR updates lack—the many tied-at-zero coordinates share indistinguishable ranks—so it lands near the bottom of the language table.

SCE (Wan et al., 2025). SCE is a matrix-aware, variance-driven merge: it optionally keeps the top fraction of coordinates by cross-expert variance, weights each expert by the mean square of its task vector (an energy weight), applies TIES-style majority-sign erasure, and forms the sign-agreeing weighted average. In our configuration the element-selection step is disabled, so its distinctive component is the energy-based per-expert weighting plus sign election. Selecting high-variance directions and weighting by update energy is meant to emphasise discriminative parameters, but the energy weighting again favours high-norm experts, reproducing the supply-determined imbalance DESA targets.

RAM (Yuan et al., 2026). RAM is designed for RL experts. On the flattened task vectors, it thresholds out near-zero entries ($|\delta| > \epsilon$, default 10^{-5}), then treats "unique" coordinates (touched by a single expert) and "overlapping" coordinates (touched by several) differently: unique entries are kept as-is, while overlapping ones are averaged over their contributing experts, before adding back to the base. This coordinate-partition rule aims to preserve expert-specific edits while diluting only genuine conflicts. It exploits the sparsity of RL updates directly, but still allocates budget coordinate-wise rather than reasoning about each expert's delivered spectral energy.

Linear average. The plain weight-space average $\boldsymbol{\theta} = \frac{1}{N}\sum_i \boldsymbol{\theta}_i$, implemented as the normalised linear operator with equal weights. Equivalently, it is Task Arithmetic with the delta sum divided by N, applying a uniform $1/N$ shrinkage to the combined update instead of leaving it at full scale. It reads no weight-internal signal and performs no interference handling, making it the most basic model-soup baseline. The uniform shrinkage tames the norm blow-up that hurts plain Task Arithmetic, but it still gives every expert an equal, energy-blind budget regardless of how far each moved from the base.

Karcher mean (Wang et al., 2026). Karcher mean replaces the Euclidean average with the Riemannian (geodesic) mean on a hypersphere: each weight tensor is split into a magnitude and a unit direction, the directions are averaged along the manifold by an iterative log/exp fixed-point update (default 10 iterations, tolerance 10^{-5}), and the result is rescaled by the weighted average of the original norms. It treats weights as points on a sphere rather than in flat space, handling direction and scale separately. It carries no notion of a shared base or per-expert budget, and performs poorly on axes such as embodied control that need a large delivered share.

MultiSLERP. A barycentric generalisation of spherical linear interpolation to N experts. Working relative to the base, it projects each task vector onto the unit hypersphere, builds their weighted Euclidean mean as a tangent-space anchor, interpolates in that tangent space, and maps the result back via the exponential map, finally rescaling by the weighted average of the input norms. Like SLERP it preserves angular geometry rather than averaging coordinates, but it requires the weighted sum to be non-degenerate. As a purely geometric interpolator with no interference or budget model, it underperforms the spectral methods on both pools.

Model Stock (Jang et al., 2024). Model Stock uses the geometry of the fine-tuned checkpoints to pick an interpolation ratio between their centre and the base. From the pairwise angle θ among the experts' offsets from the base it computes $t = N\cos\theta/(1+(N-1)\cos\theta)$ and returns $t\,\bar{\boldsymbol{w}}+(1-t)\boldsymbol{\theta}_0$, where $\bar{\boldsymbol{w}}$ is the mean of the experts. Tightly aligned experts pull the merge toward their average, while poorly aligned ones pull it back toward the base; it requires at least three models. Because RLVR experts are almost mutually orthogonal, t stays near the base, leaving recovery close to zero.

B.6 Hardware and Runtime

Expert training, merging, and evaluation are run on nodes equipped with NVIDIA A100-80GB GPUs. Merging itself is inexpensive: a single DESA merge is a one-shot spectral computation. Evaluation dominates the cost; a full core sweep over the language pool completes in roughly 40 GPU-hours on eight A100-80GB GPUs.

C More Experimental Results

C.1 Sensitivity to Rank and Global Scaling

Sensitivity to the global scaling coefficient. Table 9 shows that DESA is only mildly sensitive to the global scaling coefficient. Across the tested range, the macro average remains within a narrow band. This indicates that DESA's gains mainly come from expert-level spectral allocation, while the global merge strength plays a comparatively secondary role.

Sensitivity to Rank. We ablate the per-expert truncation rank by scaling the default TSV-style choice $k = \lfloor r/N \rfloor$. As shown in Table 8, DESA is fairly stable across a wide range of lower-rank settings: the macro average remains close to the default 68.62% from $0.875\times$ down to $0.25\times$ rank, and drops more clearly only at $0.1\times$. This suggests that DESA does not depend critically on a finely tuned rank choice, while the default TSV-style rule remains a strong operating point.

Table 8: Sensitivity of DESA to the per-expert truncation rank on the language pool.

$k/\lfloor r/N \rfloor$	1.0	0.875	0.75	0.5	0.25	0.1
Average (%)	68.618	68.180	67.593	67.800	67.907	67.205

Table 9: Ablation of the global scaling coefficient s_c in DESA.

s_c	1.00	1.20	**1.25**	1.28	1.30	1.32	1.38
Average (%)	68.5583	68.3077	**68.6180**	68.0686	68.3598	67.9180	68.0557

C.2 Additional Per-Domain Results

Tables 10 to 12 give per-benchmark accuracy for both expert pools, complementing the domain-level aggregates in Tables 1 and 2. The finer-grained results are consistent with the main-text picture: DESA's improvements are not concentrated on a single benchmark, but broadly track the same domain-level trends discussed in Section 5.

Table 10: Per-benchmark accuracy for the language pool on the tabular reasoning, tool-use, and embodied-control domains. *Oracle* is diagonal routing (each axis served by its own expert). Underlined is the best value on each benchmark (excluding the base model and *Oracle*). The mathematics and medical domains are reported in Table 11.

Method	**Table**							**Tool**	**Embodied**
	WTQ	**HiTab**	**TabMWP**	**TabMCQ**	**FeTaQA**	**TabFact**	**PubHealthTab**	**BFCL v3**	**ALFWorld**
SFT Base	46.10	54.00	76.90	68.80	35.66	80.30	81.84	47.46	1.50
Oracle	73.50	72.90	94.00	85.50	44.71	88.90	87.76	61.06	96.67
Average	67.00	64.80	91.80	77.30	38.06	86.00	88.03	50.64	62.83
TA	71.00	74.00	96.70	0.90	42.64	79.60	86.97	39.42	94.83
TIES	73.00	71.50	93.60	27.80	40.42	87.00	88.82	51.18	91.83
DARE	73.00	74.10	97.40	1.00	42.35	80.50	85.92	39.48	94.67
DARE-TIES	72.00	73.20	95.80	0.70	42.55	79.30	85.39	37.17	94.50
Breadcrumbs	71.00	73.10	95.90	0.90	42.33	79.70	86.58	39.81	93.67
DELLA	71.30	72.90	96.00	0.90	42.42	78.50	85.39	35.99	94.33
Model Stock	46.40	53.20	77.10	72.00	35.47	82.50	83.55	45.81	3.67
MultiSLERP	63.40	62.80	87.70	77.60	37.76	86.30	86.84	50.36	49.83
SCE	72.20	73.00	93.10	28.20	40.63	87.70	90.13	50.76	91.83
TSV	72.00	73.80	95.70	36.10	42.64	86.10	90.26	55.52	90.33
Iso-C	71.90	72.80	95.70	61.90	42.00	85.60	88.68	56.68	90.83
Karcher mean	67.00	64.50	89.00	79.00	38.52	84.80	88.16	51.86	49.00
RAM	70.40	72.00	93.70	38.60	39.92	88.20	90.53	51.74	91.33
RAM+	72.40	71.50	92.60	38.10	40.32	86.70	89.21	52.09	91.33
ISO-Merger	75.60	73.50	96.70	10.00	41.46	85.50	89.08	48.29	92.83
DESA (ours)	71.20	73.70	96.00	61.80	44.09	86.80	87.50	57.06	95.00

Table 11: Per-benchmark accuracy for the language pool (mathematics and medical domains).

Method	Math				Medical			
	AIME	AMC	MATH-500	OlympiadBench	MMLU-Pro-Health	MedMCQA	MedQA	PubMedQA
SFT Base	8.02	51.56	72.80	39.17	56.85	55.99	63.55	66.10
Oracle	14.58	68.83	79.00	45.99	62.71	60.98	71.33	74.00
Average	10.10	56.09	77.80	42.73	59.78	58.26	66.22	70.10
Task Arithmetic	11.98	60.23	71.20	31.16	63.94	61.39	72.43	75.00
TIES	12.50	63.44	79.20	42.43	62.96	60.98	71.41	72.80
DARE	11.67	59.84	70.00	30.56	62.84	61.08	71.96	74.60
DARE-TIES	11.56	59.14	65.40	28.78	63.57	61.27	72.90	74.90
Breadcrumbs	13.02	59.92	68.80	31.16	63.81	61.44	72.74	74.70
DELLA	10.52	58.67	66.60	28.78	63.45	61.25	72.66	75.00
Model Stock	6.25	52.19	73.40	39.32	55.13	55.92	64.65	67.20
MultiSLERP	9.06	57.34	76.80	42.28	61.00	57.92	65.04	68.40
SCE	12.50	64.14	80.00	42.14	63.08	61.46	71.64	74.10
TSV	12.40	62.58	79.60	43.47	61.98	60.70	70.54	72.90
Iso-C	10.73	61.56	79.00	42.43	61.49	60.41	69.05	72.10
Karcher mean	8.96	54.92	77.00	39.76	57.82	58.55	67.56	69.40
RAM	11.67	63.67	80.80	41.39	61.74	60.17	70.46	74.20
RAM+	12.29	61.95	79.80	41.99	61.12	60.55	71.64	74.00
ISO-Merger	13.85	63.44	76.80	39.47	63.20	61.94	73.21	73.60
DESA (ours)	11.98	61.33	80.60	45.40	62.35	60.32	70.38	74.00

Table 12: Per-benchmark accuracy for the vision-language pool.

Method	Spatial			VidObjLoc		Temporal			Math
	Maze	SPAR-Bench	MMSI-Bench	STVG	GOT-10k	ActivityNet	Charades-STA	ANet-RTL	Geometry3K
SFT Base	55.00	38.81	28.60	27.34	47.23	21.14	55.18	19.50	66.06
Oracle	70.14	40.23	30.70	39.17	78.91	41.77	63.31	35.11	73.38
Average	61.23	39.73	29.30	34.14	52.78	33.41	60.28	31.03	68.72
Task Arithmetic	70.19	41.11	29.80	39.11	78.85	39.22	61.46	29.90	71.88
TIES	63.53	40.42	28.70	38.75	78.87	41.01	63.00	33.66	69.72
DARE	70.44	41.49	29.70	39.05	78.85	39.21	61.44	29.61	71.05
DARE-TIES	69.73	41.67	29.60	38.93	78.81	38.96	61.23	29.56	72.88
Breadcrumbs	69.98	41.44	29.00	38.89	78.79	38.66	60.93	29.07	70.22
DELLA	69.83	41.59	28.40	39.02	78.80	38.53	60.76	28.79	70.22
Model Stock	55.68	38.17	28.20	29.19	48.42	22.43	55.95	20.82	65.72
MultiSLERP	62.68	39.87	29.80	34.47	53.96	34.32	60.56	32.30	68.05
SCE	63.33	40.58	29.60	39.14	78.91	41.01	63.03	33.70	70.72
TSV	67.18	40.42	29.60	37.19	74.03	40.92	63.45	34.74	69.72
Iso-C	66.93	40.78	29.90	37.41	70.69	41.19	63.57	34.29	71.21
Karcher mean	62.23	40.12	29.30	35.02	57.09	35.38	61.07	33.29	68.72
RAM	63.93	40.61	29.60	38.87	78.57	41.04	63.13	33.79	71.55
RAM+	64.28	40.60	29.20	38.68	78.57	41.01	63.14	33.59	71.21
ISO-Merger	67.83	41.23	28.60	39.19	79.01	40.85	62.92	32.80	69.88
DESA (ours)	76.59	41.84	30.40	38.61	78.03	41.54	63.37	34.31	75.04

D Limitations and Future Work

DESA is evaluated on RLVR expert pools derived from a shared base model, which aligns with a standard and practically relevant merging setting. An important direction for future work is to examine how the method behaves under broader conditions, such as larger expert collections or more heterogeneous expert pools. It would also be valuable to develop more adaptive allocation mechanisms that better reflect layer-specific differences in budget sensitivity, and to explore the integration of DESA with lightweight post-merge refinement or routing strategies to further improve robustness and generalization.