BALANCED-KNOTS: TASK-BALANCED SUBSPACE SELECTION FOR FIXED-RANK LORA MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-hoc LoRA merging can replace a growing adapter library with one static model, but under a fixed deployment rank, every expert must compete for the same limited shared space. Existing shared-SVD methods optimize aggregate reconstruction, which can preserve substantial total energy while severely underrepresenting individual tasks. We identify this as an *aggregate-diagnostic blind spot* and show that discarded task components create an *irrecoverable projection bottleneck*: no subsequent merger restricted to the selected space can reconstruct them. We introduce task-wise projection coverage to quantify per-expert representational availability and propose Balanced-KnOTS, which removes norm-induced task weighting and optimizes a mean–soft-min objective to protect the coverage lower tail before applying the same TIES procedure in the retained coordinates. On 20 independently trained Qwen3-8B security-task LoRA experts consolidated into a single rank-16 adapter, Balanced-KnOTS raises worst-task coverage from 32.84% to 56.98% while leaving mean coverage essentially unchanged. Relative to equal-rank Joint KnOTS-TIES, it improves five of six held-out task metrics, increasing mean expert recovery from 82.80% to 87.55% and worst-task recovery from 54.12% to 63.66%. These results show that fixed-rank consolidation requires controlling how shared capacity is allocated across experts, not only how retained directions are merged.

1 INTRODUCTION

Low-rank adaptation (LoRA) enables a single foundation model to be specialized into many lightweight task experts Hu et al. (2022). In practice, however, a growing adapter library fragments deployment, requiring systems to store, route, and serve multiple experts Chen et al. (2024); Zhao et al. (2024), or consolidate them into a single model. Post-hoc model merging provides the latter without joint expert training Wortsman et al. (2022); Ilharco et al. (2023); Yadav et al. (2023), and recent structured methods extend this paradigm to LoRA adapters and low-rank subspaces Stoica et al. (2025a); Panariello et al. (2026); Jeong et al. (2026b). We study the strict deployment setting in which independently trained adapters must be consolidated into one static LoRA whose rank, storage, and inference cost remain fixed as the number of experts grows. The central challenge is therefore to represent updates from different tasks within a shared subspace of fixed capacity.

A common structured-merging paradigm first constructs a shared singular-vector space and then resolves conflicts or assigns weights within that space. KnOTS, for example, jointly decomposes LoRA updates so that an existing merging rule can operate in aligned coordinates Stoica et al. (2025a). When the shared representation is restricted to rank R, joint SVD selects the subspace with the largest aggregate reconstruction energy. Although this provides a natural global approximation objective, it does not guarantee that each task receives adequate representation: high-energy or repeatedly reinforced directions may dominate the limited basis, excluding low-energy or task-specific directions. Fixed-capacity consolidation therefore requires not only preserving aggregate information, but also allocating the limited shared capacity across tasks. Figure 1 summarizes the fixed-rank LoRA consolidation pipeline and the two limitations studied in this work.

To measure how well the shared space represents each expert, we introduce *task-wise projection coverage*. With the shared rank fixed at $R = 16$, expanding a nested collection from 4 to 20 experts reduces mean coverage from 77.45% to 62.69%, whereas worst-task coverage collapses

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.

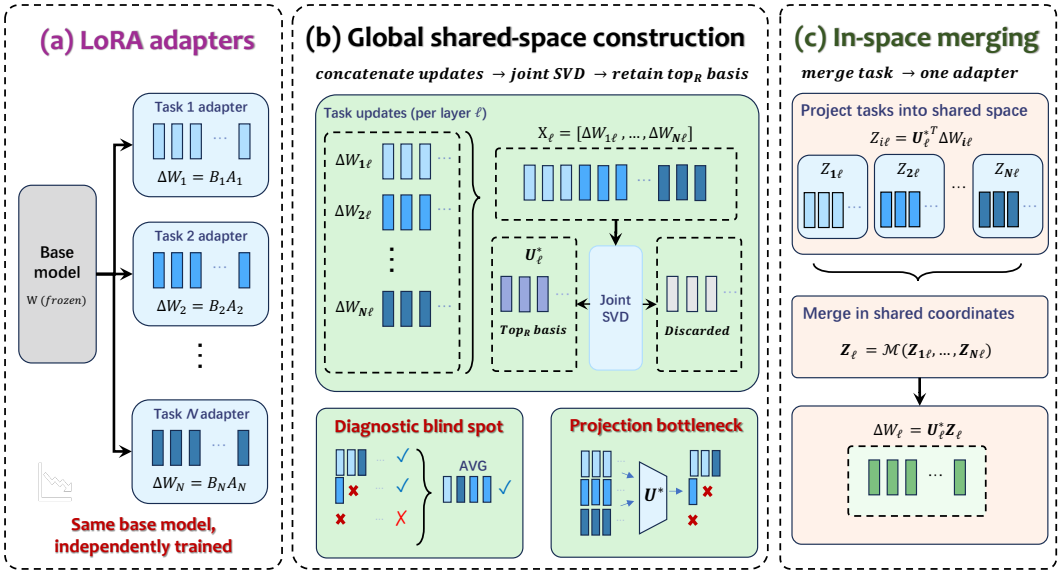

Figure 1: Overview of fixed-rank post-hoc LoRA consolidation. (a) Task-specific LoRA adapters are trained from the same base model. (b) Their updates are aggregated to construct a global rank-R shared space. Aggregate summaries may exhibit an *aggregate-diagnostic blind spot*, while task components excluded from the retained basis create an *irrecoverable projection bottleneck*. (c) The projected task updates are merged within the selected space to produce one static adapter.

from 74.38% to 32.84%. This result shows that a moderate change in an aggregate statistic can conceal severe information loss for individual tasks called as the *aggregate-diagnostic blind spot*.

This loss is not determined by the rank constraint alone. Holding both the 20-expert collection and $R = 16$ fixed, replacing joint SVD with task-normalized SVD improves worst-task coverage by 15.71 percentage points, demonstrating that how the fixed capacity is allocated across experts also matters. More fundamentally, shared-space selection creates an *irrecoverable projection bottleneck*: once a task component is discarded during rank-R projection, no subsequent merging or weighting operation restricted to the selected space can recover it. Task balance must therefore be controlled when the shared space is constructed rather than delegated entirely to the merger that follows.

Based on these insights, we propose **Balanced-KnOTS**, which balances average task coverage against the lower tail of the coverage distribution when constructing the shared space. Balanced-KnOTS then projects the task updates into the selected space and applies TIES to resolve conflicts among the retained components. The result is one static rank-R LoRA with the same deployment capacity as Joint KnOTS-TIES; no router or task-specific adapter is retained at inference time.

With 20 experts and $R = 16$, Balanced-KnOTS raises worst-task coverage from 32.84% to 56.98% while leaving mean coverage essentially unchanged. Across six heterogeneous security evaluations, it outperforms Joint KnOTS-TIES on five of the six task metrics and increases mean expert-normalized recovery from 82.80% to 87.55%.

Our main contributions are:

- We introduce task-wise projection coverage and identify the *aggregate-diagnostic blind spot*: aggregate summaries can conceal severe task-level information loss as more experts compete for a fixed-rank shared space.
- We characterize the *irrecoverable projection bottleneck*: task components discarded during shared-space construction cannot be recovered by subsequent merging or weighting operations restricted to that space.
- We propose Balanced-KnOTS, a task-balanced subspace-selection method that improves lower-tail coverage and expert-normalized performance in capacity-constrained settings without increasing deployment rank or inference-time storage.

2 Related Work

Parameter conflict resolution and adaptive aggregation. Post-hoc model merging combines models derived from a shared initialization without joint training. Task Arithmetic composes parameter differences from fine-tuned models Ilharco et al. (2022); Matena & Raffel (2022); Zhou et al. (2024); Davari & Belilovsky (2024); TIES-Merging prunes weak updates and resolves sign conflicts Yadav et al. (2023); DARE exploits parameter redundancy through random sparsification and rescaling Yu et al. (2024); and RegMean derives a merged model by preserving expert layer outputs Jin et al. (2022); Tam et al. (2023); Yadav et al. (2024); Gargiulo et al. (2025a). AdaMerging instead learns task- or layer-wise coefficients using prediction entropy on unlabeled data Yang et al. (2024); Wang et al. (2024); Tang et al. (2024). These methods determine how available task updates are selected, weighted, or aggregated. Fixed-rank consolidation introduces an earlier constraint: task updates must first be compressed into a shared low-dimensional representation. Components discarded during this stage are unavailable to any subsequent merger. We therefore study shared-space selection as an upstream problem separate from in-space aggregation.

LoRA structure, rank compression, and shared subspaces. Each LoRA adapter parameterizes a low-rank update matrix Hu et al. (2021), while combining multiple adapters can cause the resulting rank to grow with the number of experts Zhao et al. (2025). LoRA-aware merging must therefore align task updates and determine which directions to retain Stoica et al. (2025b). KnOTS constructs a shared SVD coordinate system in which existing rules such as Task Arithmetic or TIES can merge aligned task coordinates Gargiulo et al. (2025a). When this space is truncated to rank R, however, joint SVD minimizes aggregate reconstruction error without controlling how that error is distributed across tasks. Related spectral methods manipulate different structures within task updates. LoRA-LEGO clusters rank-wise minimal semantic units and assembles their centroids into a target-rank LoRA update Zhao et al. (2025); TSV-Merge compresses task matrices and reduces interference by whitening correlated task singular vectors Gargiulo et al. (2025a); and Iso-C and Iso-CTS respectively flatten the merged singular-value spectrum and augment a common subspace with task-specific directions Marczak et al. (2025b). These methods primarily characterize shared-space quality through aggregate reconstruction, clustering, or spectral properties. Our concern is complementary: when the shared rank is fixed, which tasks bear the compression error, and whether some have already lost most of their updates before aggregation begins.

Task preferences and multi-objective merging. Preference-aware merging learns task trade-offs along a Pareto front Chen & Kwok (2024); Li et al. (2025). TARA further performs preference-conditioned weighting of LoRA directions while preserving subspace coverage and mitigating directional anisotropy Jeong et al. (2026a). In its shared-basis variant, TARA reweights directions already represented in the shared space, whereas Balanced-KnOTS selects which directions to retain under a fixed-rank constraint. They therefore address task balance through in-space weighting and upstream subspace selection, respectively.

3 Problem Formulation

Post-hoc model merging consolidates fine-tuned models derived from a shared initialization without joint multi-task retraining Ilharco et al. (2023); Yadav et al. (2023). Recent work extends this setting to independently trained LoRA adapters Stoica et al. (2025a); Jeong et al. (2026b). Following this established problem setting, given N LoRA adapters derived from the same base model, we aim to consolidate them into a single static adapter with a fixed deployment rank R. Under the standard LoRA parameterization Hu et al. (2022), the update learned by task expert i at module ℓ is

$$\Delta W_{i\ell} = B_{i\ell} A_{i\ell}, \qquad \operatorname{rank}(\Delta W_{i\ell}) \le r_i, \tag{1}$$

where $B_{i\ell} \in \mathbb{R}^{d_\ell \times r_i}$ and $A_{i\ell} \in \mathbb{R}^{r_i \times m_\ell}$. All adapters modify the same base parameters and are trained independently on their respective tasks.

The output is a single adapter whose update at each module has the form

$$\Delta W_\ell^{\text{merge}} = U_\ell Z_\ell, \qquad U_\ell^\top U_\ell = I, \qquad \operatorname{rank}(\Delta W_\ell^{\text{merge}}) \le R, \tag{2}$$

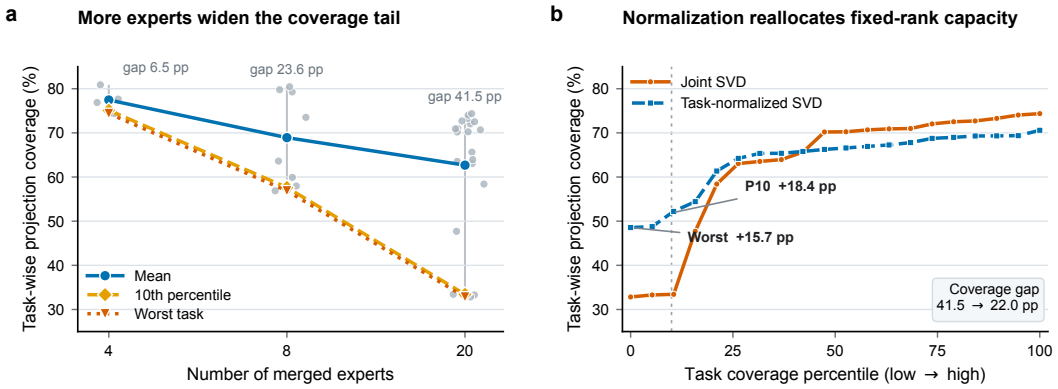

Figure 2: Task-wise projection coverage at shared rank $R = 16$. (a) Increasing the number of experts moderately reduces mean coverage but sharply degrades the lower tail. (b) For 20 experts, task normalization improves lower-tail coverage under the same rank budget.

where $U_\ell \in \mathbb{R}^{d_\ell \times R}$ defines the shared left subspace and $Z_\ell \in \mathbb{R}^{R \times m_\ell}$ contains the final merged coordinates. The collection $\{\Delta W_\ell^{\text{merge}}\}_\ell$ constitutes one static merged adapter. The shared rank R is a deployment budget and does not grow with the number of task experts.

We consider parameter-only consolidation: the method operates on the learned LoRA parameters and does not retrain the task experts, consistent with the post-hoc merging paradigm Yadav et al. (2023); Stoica et al. (2025a). The output is neither an ensemble nor a routing system and does not retain task-specific adapters at deployment. As summarized in Fig. 1, shared-space methods first construct $\mathcal{U} = \{U_\ell\}_\ell$ from the task updates and then combine or reweight the projected task coordinates to produce the single deployed matrix Z_ℓ Stoica et al. (2025a); Jeong et al. (2026b). This decomposition also separates the two issues studied in the next section: aggregate summaries may conceal which tasks are poorly represented, and components discarded during shared-space construction are unavailable to all subsequent in-space merging operations. Appendix H summarizes the notation used throughout the paper.

4 Task-wise Coverage and the Projection Bottleneck

The fixed-rank pipeline in Section 3 first selects a shared subspace and then merges task coordinates within it. We show that aggregate diagnostics can conceal task-level omission and that components discarded at this first stage cannot be recovered by subsequent in-space operations.

4.1 Task-wise Projection Coverage

For the collection of shared bases $\mathcal{U} = \{U_\ell\}_\ell$, we define the projection coverage of task i as

$$C_i(\mathcal{U}) = \frac{\sum_\ell \|U_\ell U_\ell^\top \Delta W_{i\ell}\|_F^2}{\sum_\ell \|\Delta W_{i\ell}\|_F^2}. \tag{3}$$

Here, energy is aggregated across modules before taking the ratio, so $C_i \in [0, 1]$ measures the fraction of task i's update representable in the selected spaces. We report its mean, 10th percentile, minimum, and gap $G_C = \max_i C_i - \min_i C_i$ across tasks.

4.2 Aggregate-Diagnostic Blind Spot

Retained energy, mean coverage, and LoRA-aware effective rank Jeong et al. (2026b) discard task identity. A shared space can therefore appear globally informative while omitting an individual task. Appendix A.1 formalizes this ambiguity with a two-task construction that has maximal LoRA-aware effective rank but $C_1 = 1$ and $C_2 = 0$.

As summarized in Fig. 2, this ambiguity also appears empirically at $R = 16$. In panel (a), increasing the nested expert set from 4 to 20 reduces mean coverage from 77.45% to 62.69%, but worst-task

coverage from 74.38% to 32.84%; the gap grows from 6.50 to 41.52 percentage points. Thus, the mean substantially understates lower-tail omission.

4.3 Irrecoverable Projection Bottleneck

At each module, joint SVD selects the rank-R space that maximizes $\sum_i \|U^\top \Delta W_{i\ell}\|_F^2$, or equivalently minimizes aggregate Frobenius reconstruction error. It therefore favors high-energy and mutually reinforcing directions without controlling lower-tail coverage. Appendix A.2 derives this allocation rule from the joint-SVD objective.

Panel (b) isolates allocation from rank by keeping 20 experts and $R = 16$ fixed while normalizing each nonzero module update before joint SVD. This raises worst-task coverage by 15.71 points and narrows the gap by 19.48 points. The remaining 22.04-point gap shows that scale normalization helps but does not balance the selected space.

For fixed $\mathcal{U}$, let $Z_{i\ell}$ be arbitrary task-specific coordinates used to characterize the best reconstruction available within the selected spaces. Then

$$\min_{\{Z_{i\ell}\}} \frac{\sum_\ell \|\Delta W_{i\ell} - U_\ell Z_{i\ell}\|_F^2}{\sum_\ell \|\Delta W_{i\ell}\|_F^2} = 1 - C_i(\mathcal{U}). \tag{4}$$

The $Z_{i\ell}$ above are analytical; deployment still uses the single merged Z_ℓ in Eq. equation 2. Appendix A.3 proves Eq. equation 4 by orthogonal projection. Hence, later conflict resolution or direction weighting can reweight retained components but cannot reconstruct those in $U_\ell^\perp$. This projection bottleneck shows that the task-wise coverage distribution should be explicitly controlled during shared-space selection, preventing some tasks from suffering irrecoverable information loss before in-space merging even begins.

5 Balanced-KnOTS

5.1 Overview

The analysis in Section 4 separates fixed-rank LoRA consolidation into two decisions: which task evidence remains representable in the shared space, and how the retained evidence is combined. Balanced-KnOTS addresses the first decision. For every LoRA-modified module, it selects a rank-R left subspace that balances the relative projection coverage of the task adapters. It then expresses all projected updates in canonical shared coordinates and applies the same TIES procedure as the Joint KnOTS-TIES baseline Yadav et al. (2023); Stoica et al. (2025a). The result is one static LoRA adapter of rank at most R, with no task-specific adapter or router retained at inference time.

Let $\Delta W_{i\ell} = B_{i\ell} A_{i\ell} \in \mathbb{R}^{d_\ell \times m_\ell}$ denote the effective update of task i at module ℓ, with the LoRA scale absorbed into either factor. Because modules can differ in shape and function, we construct their shared spaces independently, following the module-wise structure of KnOTS. We omit ℓ below when describing one module.

5.2 Coverage-Balanced Shared-Space Selection

For an orthonormal basis $U \in \mathbb{R}^{d \times R}$, we measure the module-level coverage of task i by

$$c_i(U) = \frac{\|U^\top \Delta W_i\|_F^2}{\|\Delta W_i\|_F^2 + \epsilon}, \qquad U^\top U = I_R. \tag{5}$$

Unlike joint SVD, which ranks directions by aggregate unnormalized energy, $c_i(U)$ measures the fraction of each task update retained by the candidate space. The model-wide diagnostic C_i in Eq. equation 3 aggregates the corresponding captured and total energies across modules; Eq. equation 5 is optimized locally because every module has its own feasible subspace.

Optimizing mean coverage removes absolute update-norm weighting but can still sacrifice a poorly covered task for small gains on many well-covered tasks. We therefore combine the mean with a

differentiable soft minimum:

$$C_{\text{mean}}(U) = \frac{1}{N}\sum_{i=1}^{N} c_i(U), \tag{6}$$

$$C_{\text{softmin}}(U) = -\tau \log\left(\frac{1}{N}\sum_{i=1}^{N} \exp\left[-\frac{c_i(U)}{\tau}\right]\right), \tag{7}$$

and select the shared space by

$$U^* \in \underset{U^\top U = I_R}{\operatorname{argmax}}\ J(U), \qquad J(U) = (1-\lambda)C_{\text{mean}}(U) + \lambda C_{\text{softmin}}(U). \tag{8}$$

The mean term preserves collection-level coverage, whereas the soft-min term emphasizes tasks with low current coverage. The temperature τ controls this lower-tail emphasis and λ controls the balance between the two terms. When $\lambda = 0$, the solution is the top-R eigenspace of the mean task-normalized left covariance; we use this task-normalized SVD solution both as an ablation and as the initialization for Eq. equation 8.

5.3 Factorized Optimization

We optimize the shared space without materializing any dense $d \times m$ task update. For each task, a symmetric eigendecomposition of the positive-semidefinite Gram matrix $A_i A_i^\top$ gives a factor L_i satisfying $L_i L_i^\top = A_i A_i^\top$. Defining $F_i = B_i L_i$ yields

$$F_i F_i^\top = \Delta W_i \Delta W_i^\top, \qquad c_i(U) = \frac{\|F_i^\top U\|_F^2}{\|F_i\|_F^2 + \epsilon}. \tag{9}$$

Thus, subspace selection depends on the original adapter ranks rather than the dense input width m. We initialize U from the leading left singular vectors of the concatenated normalized factors $[F_1/\|F_1\|_F, \ldots, F_N/\|F_N\|_F]$. We then maximize Eq. equation 8 using an Adam-style tangent-projected optimizer with a QR retraction after every step, retaining the iterate with the largest objective value. This optimization uses only the learned LoRA parameters and requires neither labels nor calibration data. Appendix C gives the factor construction, gradient, optimizer, and numerical details.

5.4 Canonical Coordinates and In-Space Merging

The objective identifies a subspace but not a unique basis within it. To obtain KnOTS-compatible coordinates, we first project each task update and diagonalize the aggregate covariance inside the selected space:

$$Z_i = U^{*\top} \Delta W_i, \qquad \sum_{i=1}^{N} Z_i Z_i^\top = Q \Lambda Q^\top, \qquad \bar{U} = U^* Q, \quad \bar{Z}_i = Q^\top Z_i. \tag{10}$$

This rotation orders the in-space axes by their unnormalized projected energy without changing the selected subspace, task coverage, or projected updates: $\bar{U}\bar{Z}_i = U^* U^{*\top} \Delta W_i$.

We apply TIES to these shared-coordinate tensors and reconstruct the merged module update:

$$Z_{\text{merge}} = \text{TIES}(\bar{Z}_1, \ldots, \bar{Z}_N), \qquad \Delta W_{\text{merge}} = \bar{U} Z_{\text{merge}}, \qquad \text{rank}(\Delta W_{\text{merge}}) \le R. \tag{11}$$

Balanced-KnOTS changes only shared-space selection: it retains the TIES procedure and hyperparameters of Joint KnOTS-TIES, together with the same output-rank budget and inference cost. With $K = \sum_i r_i$, subspace optimization requires $O(dK + dR)$ memory and $O(dRK)$ computation per step; Appendix C provides the phase-wise complexity analysis.

6 Experiments

6.1 Experimental Setup

Setting. We study fixed-rank consolidation on Qwen3-8B using 20 rank-16 LoRA experts independently trained from the same frozen checkpoint. We merge all experts into one static rank-16

Table 1: Twenty-expert consolidation on six held-out security tasks.

Method	Rep. size	Held-out task performance						ER (%)	
		CWE Acc.	CVSS Comp. Acc.	ATT&CK Acc.	Code Acc.	Extract. F1	Response R-L	Mean	Worst
Base	–	0.506	0.000	0.234	0.462	–	0.118	–	–
Single experts	r=16 each	0.719	0.865	0.579	0.468	0.077	0.333	100.00	100.00
Task Arithmetic	w=320	0.558	0.490	0.315	0.457	0.053	0.275	73.11	54.43
TIES	dense	0.559	0.153	0.309	0.369	0.043	0.265	60.48	17.73
Only truncation	w=320	0.565	0.557	0.324	0.304	0.052	0.247	67.61	55.91
TSV	w=160	<u>0.635</u>	0.540	<u>0.457</u>	0.463	0.053	**0.325**	82.57	<u>62.41</u>
Iso-C	w=320	0.615	0.206	0.349	0.460	**0.059**	0.201	67.51	23.84
Iso-CTS	w=320	0.586	0.003	0.309	<u>0.464</u>	<u>0.058</u>	<u>0.322</u>	67.92	0.39
Joint KnOTS-TIES†	r=16	0.603	<u>0.801</u>	<u>0.457</u>	0.459	0.041	0.296	<u>82.80</u>	54.12
Balanced-KnOTS†	r=16	**0.657**	**0.832**	**0.497**	**0.467**	0.049	0.294	**87.55**	**63.66**

adapter and evaluate held-out splits of six prespecified security tasks: CWE mapping, CVSS vector prediction, ATT&CK technique identification, code-vulnerability detection, vulnerability extraction, and incident-response generation. Data sources, split construction, and expert training are documented in Appendix D; task formats and merge configurations are reported in Appendix E.

Comparisons. We compare Balanced-KnOTS with Task Arithmetic Ilharco et al. (2023), TIES-Merging Yadav et al. (2023), TSV-Merge and its truncation-only control Gargiulo et al. (2025b), Iso-C, Iso-CTS Marczak et al. (2025a), and Joint KnOTS-TIES Stoica et al. (2025a). The controlled comparison is between Joint KnOTS-TIES and Balanced-KnOTS: both use the same experts, output rank, TIES procedure, retained density, and evaluation-time adapter scale, differing only in shared-space selection. Task-normalized KnOTS is evaluated separately as a component ablation.

Metrics and protocol. All methods use greedy decoding in non-thinking mode, and each task is scored with its native metric. To aggregate performance across heterogeneous metrics, we define expert recovery as

$$\mathrm{ER}_i = 100 \times \frac{M_i^{\mathrm{merge}}}{M_i^{\mathrm{expert}}}, \tag{12}$$

where M_i^{merge} and M_i^{expert} are measured on the same task and evaluation split. We report the mean and minimum ER across tasks; values above 100% indicate that the merged model exceeds the corresponding single expert. We also report the mean, 10th percentile, minimum, and max–min gap of task-wise projection coverage.

6.2 Main Results

As summarized in Table 1, Balanced-KnOTS provides the strongest overall task retention. Relative to rank-16 Joint KnOTS-TIES, it raises mean ER by 4.75 points and worst ER by 9.54 points, reaching 87.55% and 63.66%, respectively. It also improves five of the six task metrics, with the largest gains on CWE, ATT&CK, and CVSS; response ROUGE-L decreases by 0.24 points. The larger improvement in worst ER than in mean ER is consistent with the method's lower-tail objective.

The remaining baselines exhibit more task-specific strengths: TSV performs best on response generation, while Iso-C performs best on extraction. Neither translates these isolated gains into comparable aggregate retention. Balanced-KnOTS achieves the highest mean and worst ER, indicating that its main advantage lies in preserving performance more evenly across heterogeneous tasks rather than maximizing a single task score.

The geometric results support this interpretation. At $N = 20$ and $R = 16$, Balanced-KnOTS raises worst-task coverage by 24.14 points and reduces the coverage gap from 41.52 to 7.61 points, while mean coverage changes by only -0.38 points. This large lower-tail correction at nearly unchanged mean coverage indicates a redistribution of the fixed rank budget toward underrepresented experts.

The response-generation decline nevertheless cautions that a better coverage distribution need not improve every downstream metric. Section 6.3 next uses ablation experiments to isolate the contributions of task normalization and lower-tail balancing, while Section 6.4 further examines the relationship between coverage improvements and downstream task performance.

6.3 Ablation Studies

We ablate the two design choices in Balanced-KnOTS—task normalization and the soft-min lower-tail objective—in the 20-expert, rank-16 setting. All variants use the same experts, output rank, and TIES configuration; only shared-space selection changes.

Component contributions. Table 2 compares joint SVD, task-normalized SVD, and Balanced-KnOTS. Task-normalized SVD removes update-norm weighting from shared-space selection, while Balanced-KnOTS additionally applies the soft-min objective to prioritize tasks with low coverage.

Task normalization corrects a clear geometric bias: relative to joint SVD, it raises worst-task coverage by 15.71 points and nearly halves the coverage gap. This geometric improvement does not carry over to the worst downstream task, as worst ER decreases from 54.12% to 45.88%. Normalization therefore removes scale bias from space selection but remains insufficient as a task-balancing objective.

Table 2: Balanced-KnOTS ablation at rank $R = 16$ with 20 experts. Gap is in pp; all other entries are percentages.

Variant	Coverage				ER	
	Mean ↑	P10 ↑	Worst ↑	Gap ↓	Mean ↑	Worst ↑
Joint SVD	62.69	33.42	32.84	41.52	82.80	54.12
+ Task norm.	**63.86**	51.87	48.55	22.04	83.49	45.88
+ Tail obj.	62.31	**58.97**	**56.98**	**7.61**	**87.55**	**63.66**

Adding the soft-min objective produces the complementary effect. Relative to task-normalized SVD, Balanced-KnOTS raises worst-task coverage by 8.43 points and worst ER by 17.78 points, at a 1.55-point reduction in mean coverage. It consequently achieves the best lower-tail coverage and the highest mean and worst ER in the ablation. The two components thus play distinct roles: task normalization reduces norm-induced allocation bias, whereas the soft-min objective directs the remaining fixed-rank capacity toward underrepresented tasks. Appendix F.1 studies how this effect changes with capacity pressure.

6.4 Analysis and Discussion

We examine whether the lower-tail correction is localized and whether coverage gains predict downstream recovery. The first analysis covers all 20 experts and 252 LoRA-modified modules; the second uses frozen 128-example validation subsets for the four tasks shared across the expert-count–rank sweep. The full capacity sweep is reported in Appendix F.2, with additional random-subset and construction-cost analyses in Appendix G.

The lower-tail correction is distributed across the network. To determine whether the aggregate result is driven by a few anomalous modules, we use the module-level coverage in Eq. equation 5 and compute

$$\Delta c_{\min,\ell} = \min_i c_{i\ell}^{\text{Balanced}} - \min_i c_{i\ell}^{\text{Joint}}. \tag{13}$$

As summarized in Fig. 3, $\Delta c_{\min,\ell}$ is positive in all 252 modules at $N = 20$ and $R = 16$ (3.2–61.9 points), ruling out an effect driven by a few outlier layers. The correction is larger in MLP than attention projections (27.20 versus 20.32 points) and strongest in the shallow third (37.95 points, versus 27.75 and 29.64 in the middle and deep thirds). Thus, the effect is network-wide but unevenly distributed; this localization does not establish causality.

Coverage is a one-sided representational diagnostic. Task-wise projection coverage measures the fraction of a task update that remains representable after shared-space selection; ER also reflects TIES aggregation and nonlinear model behavior. As summarized in Fig. 4, 27 task–configuration pairs obtain a material coverage gain ($\Delta C > 0.1$ percentage points), but ER increases in only 15 of them, remains unchanged in four, and decreases in eight.

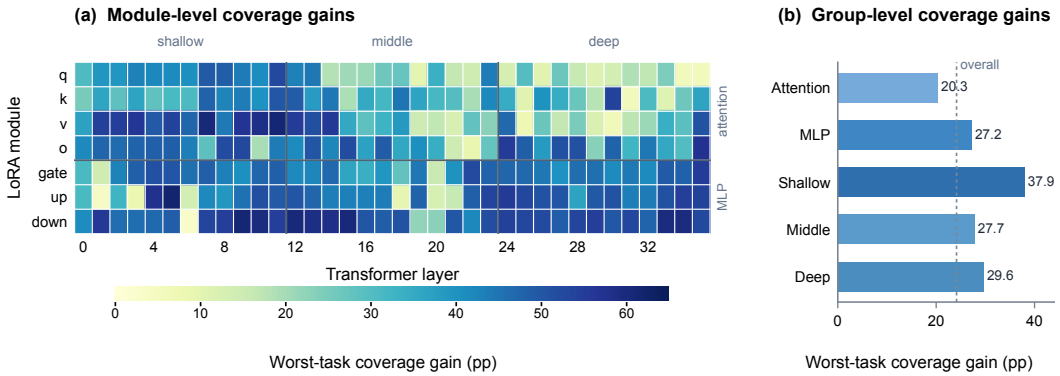

Figure 3: Layer- and module-wise lower-tail correction for 20 experts at $R = 16$. (a) Balanced-minus-Joint change in worst-task coverage for each LoRA-modified module. Separators indicate depth thirds and attention versus MLP projections. (b) Group-level changes after aggregating captured and total energies; the dashed line denotes the all-module result.

Code-vulnerability detection is the clearest counterexample: across nine configurations, coverage increases by 9.47 points on average while ER decreases by 3.07 points. The overall association is weak and uncertain (Spearman $\rho = 0.222$; task-cluster bootstrap 95% interval $[-0.227, 0.609]$), and remains descriptive given only four tasks. Coverage is measured before TIES and therefore does not capture which retained directions survive pruning, sign election, and aggregation, or how they interact through the nonlinear network. A task can consequently gain representational availability while losing useful components during in-space merging. Low coverage certifies that task components were discarded, whereas high coverage establishes only their availability.

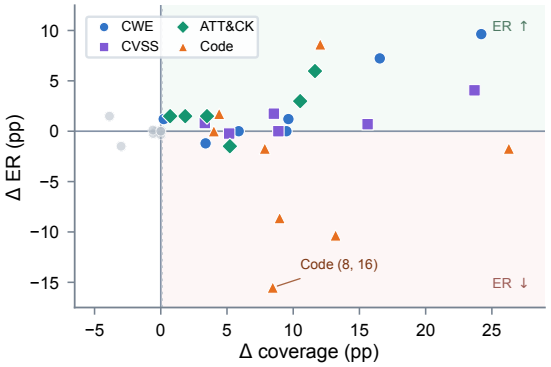

Figure 4: Balanced-minus-Joint changes in task-wise coverage and ER. Lower-right points gain coverage but lose ER.

7 Limitations

Our evaluation uses a single pool of 20 Qwen3-8B security-task adapters trained under one LoRA protocol. Broader claims require validation across domains, base models, and heterogeneous training pipelines. Balanced-KnOTS also targets tight rank budgets: its advantage narrows as the shared rank grows and may be limited when deployment permits a sufficiently wide adapter.

8 Conclusion

We study fixed-rank LoRA consolidation as two consecutive decisions: selecting a shared space and merging task coordinates within it. Aggregate diagnostics can hide task-level omission at the first stage, while components discarded there cannot be recovered by later in-space merging. Balanced-KnOTS addresses this bottleneck by removing norm-induced task weighting and protecting the lower tail of task-wise projection coverage before applying TIES. With 20 experts at rank 16, it raises worst-task coverage from 32.84% to 56.98%, mean ER from 82.80% to 87.55%, and worst ER from 54.12% to 63.66%. The gains are strongest under tight capacity, while the coverage–performance analysis shows that representational availability alone does not guarantee downstream improvement. Overall, shared-space allocation is a distinct design problem that complements conflict resolution within the selected space.

Reproducibility Statement

We provide the mathematical formulation, task-wise coverage definition, and complete Balanced-KnOTS objective in the main text. Appendix H summarizes the notation. Appendix A contains the effective-rank counterexample, the joint-SVD allocation derivation, and the projection-residual proof. Appendix C documents the factor construction, analytic gradient, tangent-projected Adam-style optimization, QR retraction, canonical coordinates, and complexity. The 20-expert pool, data sources, split controls, and unified LoRA training protocol are specified in Appendix D; evaluation formats, metrics, decoding, and merge hyperparameters are reported in Appendix E. Additional capacity, coverage–performance, robustness, and cost analyses appear in Appendices F, B, and G. Together, these sections record the assumptions, data construction, model configuration, optimization procedure, and evaluation protocol needed to reproduce the reported comparisons.

AI Use Statement

In the preparation of this paper, OpenAI Codex and ChatGPT were used as assistive tools for English drafting and translation. AI-generated material was treated as a draft rather than as empirical evidence. The authors reviewed all AI-assisted text, calculations, code, and figures against the source implementation, experimental records, underlying data, and cited literature.

All core scientific decisions and final claims were made and verified by the authors, including but not limited to:

- **Original ideas and novel viewpoints.** All innovative concepts, hypotheses, and critical analyses were independently developed and conducted by the authors.
- **Experimental results.** All code, figures, findings, and their interpretations were generated, designed, collected, and analyzed by the authors.
- **Substantive descriptions of the work.** The methodology, experimental procedures, and discussion of the research implications were written entirely by the research team.

The authors take full responsibility for the content of the paper and for all artifacts produced with the assistance of generative AI.

References

Lequn Chen, Zihao Ye, Yongji Wu, Danyang Zhuo, Luis Ceze, and Arvind Krishnamurthy. Punica: Multi-tenant lora serving. *Proceedings of Machine Learning and Systems*, 6:1–13, 2024.

Weiyu Chen and James Kwok. Pareto merging: Multi-objective optimization for preference-aware model merging. *arXiv preprint arXiv:2408.12105*, 2024.

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.

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, 2025a.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18695–18705, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Gargiulo_Task_Singular_Vectors_Reducing_Task_Interference_in_Model_Merging_CVPR_2025_paper.html.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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

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.

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

Wooseong Jeong, Wonyoung Lee, and Kuk-Jin Yoon. Preference-aligned lora merging: Preserving subspace coverage and addressing directional anisotropy. *arXiv preprint arXiv:2603.26299*, 2026a.

Wooseong Jeong, Wonyoung Lee, and Kuk-Jin Yoon. Preference-aligned LoRA merging: Preserving subspace coverage and addressing directional anisotropy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 823–835, 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Jeong_Preference-Aligned_LoRA_Merging_Preserving_Subspace_Coverage_and_Addressing_Directional_Anisotropy_CVPR_2026_paper.html.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2022.

Lu Li, Tianyu Zhang, Zhiqi Bu, Suyuchen Wang, Huan He, Jie Fu, Yonghui Wu, Jiang Bian, Yong Chen, and Yoshua Bengio. Map: Low-compute model merging with amortized pareto fronts via quadratic approximation. In *International Conference on Learning Representations*, volume 2025, pp. 65032–65064, 2025.

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. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43177–43199, 2025a. URL https://proceedings.mlr.press/v267/marczak25a.html.

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*, 2025b.

Michael S Matena and Colin Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

Aniello Panariello, Daniel Marczak, Simone Magistri, Angelo Porrello, Bartłomiej Twardowski, Andrew Bagdanov, Simone Calderara, and Joost van de Weijer. Accurate and efficient low-rank model merging in core space. *Advances in Neural Information Processing Systems*, 38:61793–61825, 2026.

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*, 2025a. URL https://openreview.net/forum?id=67X93aZHII.

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, 2025b.

Derek Tam, Mohit Bansal, and Colin Raffel. Merging by matching models in task parameter subspaces. *arXiv preprint arXiv:2312.04339*, 2023.

Anke Tang, Li Shen, Yong Luo, Nan Yin, Lefei Zhang, and Dacheng Tao. Merging multi-task models via weight-ensembling mixture of experts. *arXiv preprint arXiv:2402.00433*, 2024.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. *arXiv preprint arXiv:2405.07813*, 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.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1644c9af28ab7916874f6fd6228a9bcf-Abstract-Conference.html.

Prateek Yadav, Tu Vu, Jonathan Lai, Alexandra Chronopoulou, Manaal Faruqui, Mohit Bansal, and Tsendsuren Munkhdalai. What matters for model merging at scale? *arXiv preprint arXiv:2410.03617*, 2024.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nZP6NgD3QY.

Linyan Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Icml*, volume 2, pp. 21, 2024.

Ziyu Zhao, Leilei Gan, Guoyin Wang, Wangchunshu Zhou, Hongxia Yang, Kun Kuang, and Fei Wu. Loraretriever: Input-aware lora retrieval and composition for mixed tasks in the wild. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4447–4462, 2024.

Ziyu Zhao, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, Fei Wu, et al. Merging loras like playing lego: Pushing the modularity of lora to extremes through rank-wise clustering. In *International Conference on Learning Representations*, volume 2025, pp. 72896–72913, 2025.

Yuyan Zhou, Liang Song, Bingning Wang, and Weipeng Chen. Metagpt: Merging large language models using model exclusive task arithmetic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1711–1724, 2024.

A Supporting Analysis for Task-wise Projection Coverage

This appendix formalizes the three claims underlying the analysis in Section 4. First, LoRA-aware effective rank can remain maximal even when the selected space entirely omits one task. Second, joint SVD allocates a fixed rank budget by maximizing an energy-weighted sum of task coverage. Third, components removed during this allocation stage cannot be recovered by any subsequent operation restricted to the selected space. Together, these results separate aggregate spectral preservation, task-wise space allocation, and in-space utilization.

A.1 Effective Rank Does Not Guarantee Task-wise Coverage

Purpose. Effective rank characterizes the diversity of a collection of retained LoRA directions, whereas task-wise coverage records how much of each individual task update remains representable. We first show that the former cannot provide a non-trivial guarantee on the latter.

Write each adapter update as

$$\Delta W_{i\ell} = \sum_j b_{i\ell j} a_{i\ell j}^\top, \tag{14}$$

and form the projected LoRA-aware rank-one stack

$$X_\ell^{\mathrm{aware}}(U_\ell) = \left[\mathrm{vec}\big(U_\ell U_\ell^\top b_{i\ell j} a_{i\ell j}^\top\big)^\top \right]_{i,j}. \tag{15}$$

For the unprojected adapter collection, the projector is omitted. Given the singular values $\{\sigma_k\}$ of this row-stacked matrix, its entropy-based effective rank is

$$r_{\text{eff}}(X_\ell^{\text{aware}}(U_\ell)) = \exp\left(-\sum_k p_k \log p_k\right), \qquad p_k = \frac{\sigma_k^2}{\sum_j \sigma_j^2}. \tag{16}$$

Proposition 1 (Aggregate diversity can coexist with complete task omission). LoRA-aware effective rank places no non-trivial lower bound on worst-task projection coverage. In particular, a selected rank-R space can satisfy $r_{\text{eff}} = R$ while attaining $\min_i C_i = 0$.

Proof. Consider two rank-R updates with orthogonal left subspaces,

$$\Delta W_1 = \alpha U_1 V_1^\top, \qquad \Delta W_2 = \beta U_2 V_2^\top, \qquad U_1^\top U_2 = 0, \tag{17}$$

where $U_i = [u_{i1}, \ldots, u_{iR}]$ and $V_i = [v_{i1}, \ldots, v_{iR}]$ have orthonormal columns, $\alpha \gg \beta$, and both spectra are flat. The top-R joint-SVD space is $\text{span}(U_1)$. After projection, the rank-one rows contributed by task 2 vanish, whereas

$$\left\{\text{vec}(\alpha u_{1j} v_{1j}^\top)^\top\right\}_{j=1}^R \tag{18}$$

are mutually orthogonal and have equal norm. Therefore, $X^{\text{aware}}(U_1)$ has R equal non-zero singular values and $r_{\text{eff}} = R$. At the same time,

$$C_1 = 1, \qquad C_2 = 0. \tag{19}$$

Thus, even maximal aggregate effective rank does not prevent complete omission of an individual task. □

Implication. Effective rank remains useful for describing global spectral diversity, but it cannot replace a task-indexed diagnostic. This is a logical limitation of the aggregate statistic; the proposition does not claim that high effective rank empirically causes low task coverage.

A.2 Joint SVD Performs Energy-weighted Coverage Allocation

Purpose. We next identify the allocation rule implicitly implemented by joint SVD. This clarifies why an aggregate-optimal space need not protect low-energy or geometrically distinct tasks.

For module ℓ, concatenate the task updates as

$$X_\ell = [\Delta W_{1\ell}, \ldots, \Delta W_{N\ell}]. \tag{20}$$

Define the update energy and relative module-level coverage as

$$e_{i\ell} = \|\Delta W_{i\ell}\|_F^2, \qquad c_{i\ell}(U) = \frac{\|U^\top \Delta W_{i\ell}\|_F^2}{\|\Delta W_{i\ell}\|_F^2}. \tag{21}$$

Proposition 2 (Aggregate-energy allocation). The leading rank-R left singular space of X_ℓ maximizes an energy-weighted sum of task-wise relative coverage:

$$U_\ell^{\text{joint}} = \arg\max_{U^\top U = I} \sum_{i=1}^N e_{i\ell} c_{i\ell}(U). \tag{22}$$

Proof. By the variational characterization of truncated SVD,

$$U_\ell^{\text{joint}} = \arg\max_{U^\top U = I} \text{tr}(U^\top X_\ell X_\ell^\top U) \tag{23}$$

$$= \arg\max_{U^\top U = I} \sum_{i=1}^N \|U^\top \Delta W_{i\ell}\|_F^2 \tag{24}$$

$$= \arg\max_{U^\top U = I} \sum_{i=1}^N e_{i\ell} c_{i\ell}(U), \tag{25}$$

where the second equality follows from

$$X_\ell X_\ell^\top = \sum_i \Delta W_{i\ell} \Delta W_{i\ell}^\top. \tag{26}$$

This proves Eq. equation 22. □

Implication. Joint SVD is optimal for aggregate Frobenius reconstruction, not for minimum or lower-tail task coverage. Large-norm updates receive greater implicit weight, and multiple tasks occupying similar directions accumulate spectral mass in the same region. Normalizing each task update removes the first effect but not the second, because the resulting objective still optimizes an average across tasks.

A.3 Post-selection Residual Is Irrecoverable

Purpose. The previous result concerns how the shared space is selected. We finally distinguish this selection stage from the later use of directions inside the selected space.

Proposition 3 (Post-selection irrecoverability). For a fixed collection of orthonormal bases $\mathcal{U} = \{U_\ell\}_\ell$, the smallest relative reconstruction error attainable for task i by any in-space coefficients $\{Z_{i\ell}\}_\ell$ is

$$\min_{\{Z_{i\ell}\}} \frac{\sum_\ell \|\Delta W_{i\ell} - U_\ell Z_{i\ell}\|_F^2}{\sum_\ell \|\Delta W_{i\ell}\|_F^2} = 1 - C_i(\mathcal{U}). \tag{27}$$

Proof. For each module, the orthogonal projection theorem gives

$$\arg\min_Z \|\Delta W_{i\ell} - U_\ell Z\|_F^2 = U_\ell^\top \Delta W_{i\ell}. \tag{28}$$

The corresponding residual is $(I - U_\ell U_\ell^\top)\Delta W_{i\ell}$. By Pythagorean decomposition,

$$\|\Delta W_{i\ell}\|_F^2 = \|U_\ell U_\ell^\top \Delta W_{i\ell}\|_F^2 + \|(I - U_\ell U_\ell^\top)\Delta W_{i\ell}\|_F^2. \tag{29}$$

Summing over modules and normalizing by the total task-update energy yields Eq. equation 27. □

Implication. Once the shared space is fixed, conflict resolution or direction weighting can only redistribute influence among retained directions. It cannot reconstruct components in $U_\ell^\perp$. This statement concerns representational availability rather than downstream task performance: low coverage certifies missing task-update components, whereas high coverage does not guarantee that the merger will use the retained components effectively.

B Coverage–Performance Association

This appendix examines the empirical relationship between task-wise projection coverage and downstream capability recovery. The purpose is to determine the explanatory boundary of coverage, rather than to use it as a substitute for task performance. Specifically, we ask two questions: whether configurations with higher task coverage also attain higher expert-normalized recovery (ER), and whether the coverage improvement produced by Balanced-KnOTS over Joint KnOTS-TIES is accompanied by a corresponding improvement in ER.

B.1 Analysis Protocol

We pool task-level observations across the evaluated expert-count and rank configurations. We first measure the association between absolute task-wise coverage and the corresponding ER. We then form paired differences between Balanced-KnOTS and Joint KnOTS-TIES under the same task and configuration, and measure the association between ΔCoverage and ΔER. Because the relationship need not be linear, both analyses use Spearman rank correlation. We report the point estimate, its 95% confidence interval, and a two-sided p-value. The evaluated tasks form only four task clusters, so we do not draw task-subgroup conclusions from these pooled statistics.

B.2 Results

Table 3 reports the two associations. Both point estimates are weakly positive, but both confidence intervals cross zero and neither test is statistically significant. The pooled configurations therefore do not support a reliable monotonic relationship between coverage and ER, nor between their method-induced changes.

Table 3: Association between projection coverage and expert-normalized recovery.

Comparison	ρ	95% CI	p
Coverage vs. ER	0.169,6	$[-0.0376,\ 0.4814]$	0.154,5
ΔCoverage vs. ΔER	0.221,7	$[-0.2267,\ 0.6093]$	0.193,8

Spearman rank correlations are reported. Confidence intervals are at the 95% level; all tests are two-sided.

Table 4: Coverage–recovery counterexamples on the code-vulnerability task.

Experts, N	Rank, R	ΔCoverage	ΔER
8	16	8.46	−15.52
20	32	13.20	−10.34
8	32	8.97	−8.62

The weak pooled association is not merely a consequence of small numerical changes. Table 4 lists three configurations from the code-vulnerability task in which Balanced-KnOTS substantially increases coverage but decreases ER. These cases rule out interpreting coverage improvement as a sufficient condition for downstream improvement.

B.3 Interpretation

Task-wise coverage measures whether an expert update remains representable after the shared space is selected. Low coverage therefore certifies that some component of that update is unavailable to every subsequent operation restricted to the selected space. It does not determine how the retained coordinates are weighted by TIES, whether their interactions are beneficial after aggregation, or how the resulting parameter change affects the nonlinear network. Consequently, coverage is a geometric availability diagnostic rather than a standalone proxy for downstream performance.

This boundary does not alter the fixed-rank allocation result. The expert-count and rank ablations show that Balanced-KnOTS provides its largest coverage and worst-task ER advantages when many experts compete for a small shared rank, while the advantages narrow as capacity increases. The association analysis instead shows that task-level performance differences cannot be attributed to coverage differences alone. Shared-space selection and in-space utilization should therefore be evaluated as related but distinct stages of the merging pipeline.

C Optimization and Implementation Details of Balanced-KnOTS

This appendix complements Section 5 with the algebra used to avoid dense task updates, the gradient and constrained optimizer, and the complete implementation and complexity details.

C.1 Low-Rank Left-Factor Construction

For one module, let $\Delta W_i = B_i A_i$, where $B_i \in \mathbb{R}^{d\times r_i}$ and $A_i \in \mathbb{R}^{r_i\times m}$. We form the small Gram matrix $A_i A_i^\top$ in FP64 and compute its symmetric eigendecomposition

$$A_i A_i^\top = V_i \Lambda_i V_i^\top. \tag{30}$$

Let $V_{i,+}$ and $\Lambda_{i,+}$ contain the eigenvectors and eigenvalues retained above the numerical support threshold $\max(10^{-7}, 10^{-6}\lambda_{i,\max})$. We construct

$$L_i = V_{i,+}\Lambda_{i,+}^{1/2}, \qquad F_i = B_i L_i. \tag{31}$$

Here L_i is a Gram factor obtained from the eigendecomposition, not the symmetric principal square root and not a Cholesky factor. It satisfies

$$\begin{aligned} F_i F_i^\top &= B_i L_i L_i^\top B_i^\top = B_i A_i A_i^\top B_i^\top = \Delta W_i \Delta W_i^\top, \\ \|F_i\|_F^2 &= \mathrm{Tr}(F_i F_i^\top) = \|\Delta W_i\|_F^2. \end{aligned} \tag{32}$$

Consequently, both coverage and covariance–basis products can be evaluated without constructing ΔW_i:

$$c_i(U) = \frac{\|F_i^\top U\|_F^2}{\|F_i\|_F^2 + \epsilon}, \qquad G_i U = \frac{F_i(F_i^\top U)}{\|F_i\|_F^2 + \epsilon}, \quad G_i = \frac{\Delta W_i \Delta W_i^\top}{\|\Delta W_i\|_F^2 + \epsilon}. \tag{33}$$

The resulting F_i is stored in FP32 after the FP64 decomposition.

C.2 Objective Gradient

The soft-min term in Eq. equation 7 induces the normalized weights

$$p_i(U) = \frac{\exp[-c_i(U)/\tau]}{\sum_j \exp[-c_j(U)/\tau]}, \qquad w_i(U) = \frac{1-\lambda}{N} + \lambda p_i(U). \tag{34}$$

Because $\nabla_U c_i(U) = 2G_iU$, the Euclidean gradient of the balanced objective is

$$\nabla_U J(U) = 2\sum_{i=1}^{N} w_i(U) G_i U. \tag{35}$$

Lower-coverage tasks receive larger soft-min weights, while the uniform component retains pressure on the collection-wide mean.

C.3 Tangent-Projected Optimization

We initialize U with the leading R left singular vectors of

$$\left[\frac{F_1}{\|F_1\|_F}, \ldots, \frac{F_N}{\|F_N\|_F}\right], \tag{36}$$

which is the optimizer of the mean-coverage objective. Before the formal iterations, we add a deterministic, statistics-derived perturbation of magnitude 10^{-5} in the orthogonal complement of the initialized space and re-orthogonalize. The perturbation is used only to escape exact invariant-axis saddle points that can occur for synthetic or duplicated experts; it is not a random initialization.

At iteration t, let $H_t = \nabla_U J(U_t)$. We project it onto the tangent space of the Stiefel constraint:

$$\widetilde{H}_t = H_t - U_t \,\mathrm{sym}(U_t^\top H_t), \qquad \mathrm{sym}(X) = \frac{X + X^\top}{2}. \tag{37}$$

The implementation then applies element-wise Adam-style moments,

$$M_t = \beta_1 M_{t-1} + (1-\beta_1)\widetilde{H}_t, \tag{38}$$

$$V_t = \beta_2 V_{t-1} + (1-\beta_2)\widetilde{H}_t^{\odot 2}, \tag{39}$$

with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. After bias correction, an ascent step in ambient coordinates is followed by a thin-QR retraction:

$$U_{t+1} = \mathrm{qf}\left(U_t + \eta \frac{\widehat{M}_t}{\sqrt{\widehat{V}_t} + \epsilon_{\mathrm{Adam}}}\right). \tag{40}$$

The QR factor is normalized to deterministic column signs. The moment estimates are not transported between tangent spaces; hence this procedure is an Adam-style tangent-projected optimizer with QR retraction, rather than a dedicated Riemannian Adam method. We retain the iterate with the largest value of $J(U)$.

C.4 Canonicalization and Reconstruction

Given the optimized basis U^*, define $Z_i = U^{*\top}\Delta W_i$ and eigendecompose the unnormalized projected covariance:

$$\sum_i Z_i Z_i^\top = Q \Lambda Q^\top. \tag{41}$$

The rotated basis and coordinates $\bar{U} = U^*Q$ and $\bar{Z}_i = Q^\top Z_i$ preserve both the projector and each projected update:

$$\bar{U}\bar{U}^\top = U^*U^{*\top}, \qquad \bar{U}\bar{Z}_i = U^*U^{*\top}\Delta W_i. \tag{42}$$

This canonicalization provides singular-value-weighted, KnOTS-compatible coordinates before TIES. Joint and Balanced variants then share the same global pruning procedure, retained-density semantics, zero-sign majority rule, nonzero-agreement mean, and evaluation-time adapter scale. Since these quantities are evaluated on different coordinate tensors, their realized thresholds, masks, and elected signs need not coincide.

C.5 Hyperparameters and Phase-Wise Complexity

We use $\lambda = 0.5$, $\tau = 0.05$, 50 projected-optimization steps, and learning rate $\eta = 0.01$ for all expert counts and rank budgets. Coverage denominators and norm clamps use $\epsilon = 10^{-30}$, while the Adam denominator uses $\epsilon_{\text{Adam}} = 10^{-8}$. TIES uses retained density 0.2, and all compared Joint and Balanced variants use evaluation-time adapter scale 1.5. Decompositions are performed in FP64, after which factors and shared coordinates are stored in FP32.

Let $K = \sum_i r_i$ denote the total input adapter rank for one module. Constructing the task Gram matrices and their retained factors costs

$$O\left(\sum_i m r_i^2 + \sum_i r_i^3 + \sum_i d r_i^2\right). \tag{43}$$

The normalized initialization applies a thin SVD to a $d \times K$ matrix. During balanced optimization, the retained left factors and optimizer state require $O(dK + dR)$ memory; each objective step costs $O(dRK)$, and T steps cost $O(TdRK)$. This stage is independent of the dense input width m once the left factors have been constructed.

After selecting the space, each shared-coordinate tensor can be formed as

$$Z_i = (U^{*\top} B_i) A_i \tag{44}$$

at total cost $O(RK(d+m))$ across tasks. Forming the $R \times R$ in-space covariance costs $O(NR^2m)$, followed by an $O(R^3)$ eigendecomposition. TIES processes NRm coordinate values for the module and requires $O(NRm)$ peak coordinate storage when all task coordinates for that module are resident. The exported LoRA factors require $O(R(d+m))$ storage. Thus, Balanced-KnOTS avoids materializing the N dense $d \times m$ task updates, but its coordinate-merging stage still depends on m through the rank-R right factors.

D Expert-Pool Construction and Training

D.1 Pool Definition

All task experts are trained independently from the same frozen `Qwen/Qwen3-8B` checkpoint; they are not collected from heterogeneous public adapter repositories. The base weights, tokenizer, chat template, LoRA rank, and modified modules are shared across experts. This controlled construction makes the resulting module updates dimensionally compatible and isolates task training as the source of variation before post-hoc consolidation.

We fixed the 20-task pool before running the merging and rank-allocation experiments. The pool was designed to combine compatible training conditions with heterogeneous security objectives, covering categorical or identifier prediction, fixed-field structured prediction, information extraction, and open-ended analysis or response generation. Each adapter contains 43,646,976 trainable parameters and occupies approximately 175 MB when stored in FP32.

The term "20 experts" refers to 20 independently trained task adapters, not to 20 statistically independent capabilities. In particular, several SEVENLLM tasks can share an underlying security-event context while differing in their instructions and output targets. We control observable cross-split overlap as described below, but treat each frozen adapter as one task update whose evidence should remain available during merging.

D.2 Data Sources and Task Definitions

The expert pool is constructed from four fixed sources:

- **NVD 2024 CVE snapshot:** CWE mapping and CVSS v3.1 base-vector prediction. Inputs contain vulnerability descriptions but exclude the target CWE identifier and CVSS fields.
- **MITRE Enterprise ATT&CK procedure relationships:** attack-technique identification from behavioral or procedural evidence, with targets mapped to parent technique identifiers.
- **Fixed PrimeVul mirror:** binary vulnerability detection for C/C++ functions, preserving the original pair identities and split assignments.
- **Fixed SEVENLLM version:** the remaining 16 extraction, event-analysis, and response tasks. The source version is fixed at commit `1de23ce55cadc984d3f3a7b52c4035a68c6cd5b0`.

Table 5 lists the task definitions and split sizes after cleaning and global grouping. "Used train" applies the per-task budget of at most 2,048 examples. Counts refer to task records; because one security event can yield records for multiple task formulations, they should not be interpreted as counts of deduplicated independent events.

After cleaning and split construction, the 20 tasks contain 82,373 training records, 23,515 validation records, and 17,087 test records. Applying the per-task budget yields 35,111 records used for expert training. Test identities are frozen from official, temporal, or fixed internal holdouts and are never selected according to model performance.

D.3 Cross-Task Splits and Leakage Control

Before creating task files, we construct global relation groups and assign each group wholly to train, validation, or test. Grouping keys include normalized text, CVE identifiers, projects and commits, attack entities, and relations found through near-duplicate detection. Consequently, the same vulnerability, code commit, or security event cannot appear on the training side of one task and the test side of another, even when it is converted into multiple instruction templates.

For the NVD tasks, we use a temporal holdout: records published from October through November 2024 form the validation pool, and December 2024 records form the test pool. Global relation grouping can move related records into a common split when necessary. For the remaining sources, we preserve official split identities where available and otherwise apply grouped assignment with a fixed hash and random seed 42. Official test records and reserved external test groups receive priority when resolving grouping constraints.

These controls address observable entity overlap and near duplication in the constructed datasets. They cannot rule out the possibility that the base model encountered the same public security material during pretraining. Nor do they make the SEVENLLM objectives independent when they originate from a shared event context; that correlation is retained as a property of the expert pool.

D.4 Unified LoRA Training Protocol

All experts use the common configuration in Table 6. Each task contributes at most 2,048 training records; tasks below the cap use all eligible records. Sampling order and random seeds are frozen before training.

Explicit thought fields are excluded, and loss is computed only over target answer tokens. Samples longer than 4,096 tokens are excluded during training-set selection rather than truncated. For records governed by pair or relation constraints, the entire relation group is excluded if any member fails the length requirement, preserving group identity.

For each task, a fixed set of 32 validation examples is used only as a training-pipeline sanity check, not to choose a checkpoint by downstream performance. All experts use the final checkpoint after one epoch. Every training job passes save–reload verification, and its post-training loss is lower than its pre-training loss on the corresponding sanity examples. These checks verify that the adapter training and loading pipeline functions correctly; they do not replace held-out task evaluation.

Table 5: Task composition and split sizes of the 20-expert pool.

			Training		Evaluation	
ID	Task	Input → output	Available	Used	Val.	Test
NVD						
1	CWE mapping	Vulnerability description → CWE ID	13,754	2,048	7,479	4,236
2	CVSS prediction	Vulnerability description → CVSS v3.1 vector	15,922	2,048	8,454	4,780
ATT&CK						
3	ATT&CK technique identification	Behavioral evidence → parent technique ID	10,591	2,048	1,291	1,167
PrimeVul						
4	Code-vulnerability detection	C/C++ function → binary label	5,790	2,048	1,777	1,585
SEVENLLM						
5	Threat analysis	Event text → threat analysis	5,199	2,048	677	910
6	Risk assessment	Event text → risk assessment	3,208	2,048	385	480
7	Incident response	Event text → response plan	1,958	1,958	247	262
8	Defense strategy	Event text → defense recommendations	3,728	2,048	433	543
9	Malware-feature extraction	Event text → software and behavior features	3,141	2,048	407	592
10	Security-event classification	Event evidence → event category	3,402	2,048	391	528
11	Attack-strategy analysis	Event description → attack strategy and procedure	1,917	1,917	237	238
12	Attack-intent analysis	Behavioral evidence → attack objective	1,068	1,068	103	124
13	Behavioral-pattern analysis	Behavior fragment → pattern and stage	1,091	1,091	159	133
14	Event-correlation analysis	Event or entity clues → relationships	947	947	137	135
15	Impact-scope analysis	Event conditions → affected scope	1,598	1,598	192	237
16	Exploit-condition analysis	Vulnerability material → exploit conditions and causes	1,042	1,042	116	107
17	Attack-tool identification	Event text → tools and evidence	2,392	2,048	307	310
18	Attacker-information extraction	Event text → attacker information	2,663	2,048	327	345
19	Vulnerability-information extraction	Event text → vulnerability, product, and version fields	1,968	1,968	273	256
20	Security entity-relation extraction	Event text → entity relations	994	994	123	119

Note. Available denotes the number of training examples remaining after cleaning and construction of the global splits. Used denotes the number retained after applying the per-task cap of 2,048 examples.

Table 6: Training configuration shared by all task experts.

Item	Configuration
Base model	Qwen3-8B; frozen during training
LoRA rank / α / dropout	16 / 32 / 0.05
Target modules	q, k, v, o, gate, up, and down projections
Base / adapter precision	BF16 / FP32
Per-task training budget	At most 2,048 records
Training epochs	1
Maximum sequence length	4,096 tokens
Micro-batch / gradient accumulation	1 / 16
Optimizer learning rate	1×10^{-4}
Learning-rate schedule	Cosine with 5% warmup
Weight decay	0.01
Random seed	42
Loss	Autoregressive loss over valid answer tokens only
Training chat mode	`enable_thinking=False`
Selected checkpoint	Final checkpoint after the single epoch

D.5 Nested Expert Collections and Evaluated Tasks

The expert collections used in the scale study were fixed before any merging results were observed:

$$\begin{aligned}
\mathcal{S}_4 &= \{\texttt{cwe_mapping}, \texttt{cvss_prediction}, \\
&\quad \texttt{attack_technique}, \texttt{code_vulnerability}\}, \\
\mathcal{S}_8 &= \mathcal{S}_4 \cup \{\texttt{vulnerability_extraction}, \texttt{relation_extraction}, \\
&\quad \texttt{threat_analysis}, \texttt{incident_response}\}, \\
\mathcal{S}_{20} &= \mathcal{S}_8 \cup \{\texttt{risk_assessment}, \texttt{defense_strategy}, \\
&\quad \texttt{malware_features}, \texttt{event_classification}, \\
&\quad \texttt{attack_strategy}, \texttt{attack_intent}, \\
&\quad \texttt{behavioral_patterns}, \texttt{event_correlation}, \\
&\quad \texttt{impact_scope}, \texttt{vulnerability_analysis}, \\
&\quad \texttt{attack_tools}, \texttt{attacker_profile}\}.
\end{aligned}$$

$\mathcal{S}_4$ contains four label or structured-decision tasks with deterministic automatic scoring. $\mathcal{S}_8$ adds extraction and open-ended tasks to increase output and subspace heterogeneity, and $\mathcal{S}_{20}$ contains the complete pool. The memberships are held fixed across all expert-count and rank comparisons and are not adjusted using validation or test performance from any merging method.

The main experiment merges all 20 adapters but performs complete held-out test evaluation on six prespecified representatives: the four tasks in $\mathcal{S}_4$, vulnerability-information extraction, and incident response. Projection coverage is therefore available for all 20 adapters, whereas the main-table mean and worst ER summarize only these six evaluated tasks and should not be interpreted as an empirical performance lower bound over the complete 20-task pool.

E Additional Experimental Details

This appendix records the task-specific evaluation metrics, merge settings, and capacity-matching protocol used in Section 6. The construction, data splits, and training of the 20 task experts are documented separately in Appendix D.

E.1 Evaluation Tasks and Metrics

The main experiment evaluates held-out test splits of six tasks with heterogeneous output protocols. Table 7 defines the metric used for each task. JSON leaf F1 and ROUGE-L measure structural or

Table 7: Evaluation tasks, output protocols, and task-native metrics.

Task	Output protocol	Metric
CWE mapping	Categorical identifier	Accuracy
CVSS vector prediction	Eight-field structured vector	Mean component accuracy
ATT&CK technique identification	Categorical identifier	Accuracy
Code-vulnerability detection	Categorical label	Accuracy
Vulnerability extraction	JSON structure	JSON leaf F1
Incident-response generation	Open-ended text	ROUGE-L

Table 8: Fixed merge and evaluation settings.

Item	Configuration
Study design	
Main expert count / output rank	$N = 20$ / $R = 16$
Capacity study	$N \in \{4, 8, 20\}$, $R \in \{16, 32, 64\}$
Merging	
TIES retained density	0.2
Evaluation-time adapter scale	1.5
Balance coefficient λ	0.5
Soft-min temperature τ	0.05
Projected-optimization steps	50
Projected-optimization learning rate	0.01
Evaluation	
Decoding	Greedy; thinking disabled
Main-table evaluation	Complete held-out test splits
Capacity-study evaluation	Frozen 128-example validation subsets

reference overlap; neither should be interpreted as a direct measure of factual correctness or human-perceived response quality.

E.2 Merge and Evaluation Configuration

Table 8 lists the settings shared across the main and capacity-pressure studies. Joint KnOTS-TIES, task-normalized KnOTS, and Balanced-KnOTS use the same TIES rule and hyperparameters after constructing their respective shared spaces. Their realized thresholds, pruning masks, and elected signs remain data-dependent because their coordinate tensors can differ.

E.3 Capacity Matching

Joint KnOTS-TIES and Balanced-KnOTS both export one rank-16 adapter in the main comparison and therefore have matched deployment rank and inference cost. Task-normalized KnOTS uses the same output budget and serves as a shared-space ablation. The remaining transferred baselines retain their native representations: TSV uses factor width 160, Task Arithmetic, the truncation-only control, Iso-C, and Iso-CTS use factor width 320, and direct TIES produces a dense update. Here width denotes the exported factor width and should not be interpreted as an independently estimated numerical matrix rank. These methods are included to contextualize achieved performance, while conclusions about equal-budget improvement are based on the Joint-versus-Balanced comparison.

F Capacity Pressure Across Expert Counts and Rank Budgets

This appendix examines how the effect of balanced shared-space allocation changes with the number of experts and the available rank budget. We evaluate the fixed nested collections $\mathcal{S}_4 \subset \mathcal{S}_8 \subset \mathcal{S}_{20}$ at $R \in \{16, 32, 64\}$. Coverage is computed over every expert in the active collection. ER is computed

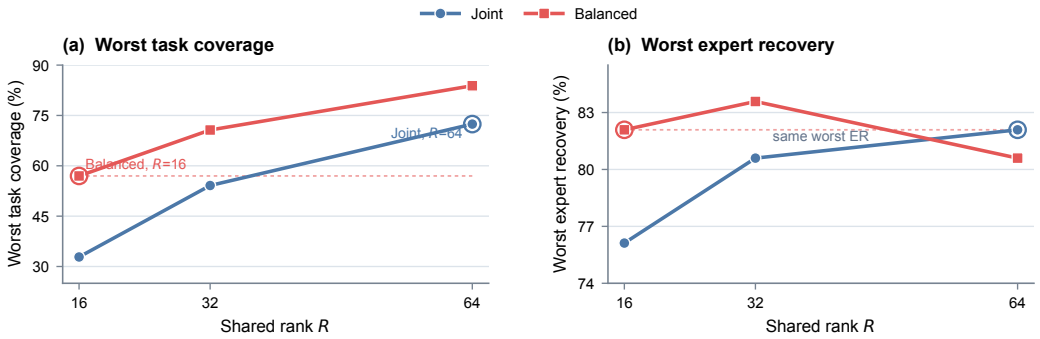

Figure 5: Rank efficiency with 20 experts on the frozen four-task validation subsets. Dashed lines mark Balanced-KnOTS at rank 16. (a) Balanced rank 16 exceeds Joint rank 32 in worst-task projection coverage. (b) Among the sampled ranks, Joint first matches the worst ER of Balanced rank 16 at rank 64. The comparison covers only $R \in \{16, 32, 64\}$ and does not establish a general compression ratio.

over the four evaluation tasks shared by all three collections, using the same frozen 128-example validation subset per task. These ER values characterize the capacity sweep and should not be compared numerically with the six-task test results in Table 1.

F.1 Rank-Efficiency View

Figure 5 provides a deployment-oriented view of the $N = 20$ slice. At rank 16, Balanced-KnOTS attains 56.98% worst-task coverage and 82.09% worst ER. Joint KnOTS at rank 32 remains below both values, at 54.13% and 80.60%, respectively. Among the sampled ranks, Joint first matches the worst ER of Balanced rank 16 at rank 64, where it reaches 82.09% ER and 72.42% coverage.

This comparison shows that better allocation can defer rank expansion in the present setting. It should not be interpreted as a general compression ratio: the observation is restricted to three sampled ranks, one expert pool, and four validation tasks. It also does not imply monotonic downstream behavior for Balanced-KnOTS, whose geometric gains and ER changes need not move together.

F.2 Full Expert-Count–Rank Grid

Figure 6 reports the complete $N \times R$ grid. At fixed $R = 16$, the Balanced-minus-Joint gain in worst-task coverage increases from 2.63 points with four experts to 9.63 points with eight experts and 24.14 points with 20 experts. The corresponding worst-ER gains are 1.49, 1.49, and 5.97 points. Within this nested sequence, balanced allocation becomes more consequential as additional experts compete for the same shared rank.

The complementary $N = 20$ slice shows how the effect changes as capacity increases. When R grows from 16 to 32 and 64, the worst-coverage gain decreases from 24.14 to 16.54 and 11.43 points. The worst-ER difference changes from $+5.97$ to $+2.99$ and then -1.49 points. Thus, Balanced-KnOTS produces a more balanced projection throughout the tested 20-expert sweep, but the geometric gain does not imply a uniform downstream advantage once capacity pressure is relaxed.

G Additional Robustness and Cost Analyses

This appendix complements the capacity analysis in Appendix F and the prediction-level diagnostics in Appendix B. All analyses use the frozen adapters, merge configurations, decoding protocol, and metrics described in Appendix E.

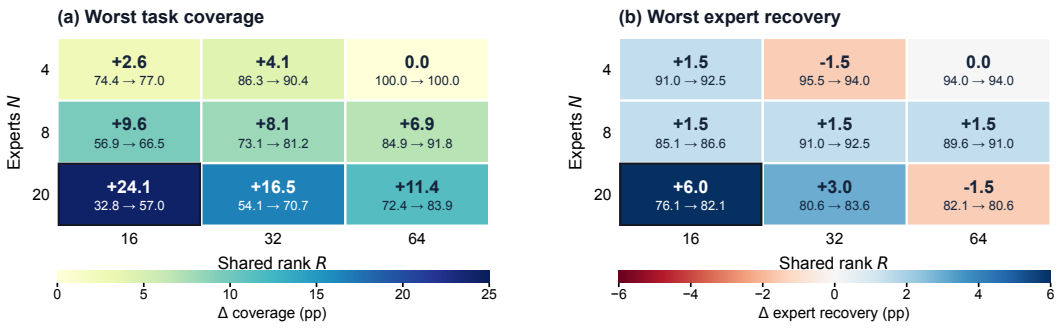

Figure 6: Balanced-minus-Joint changes across expert counts and shared-rank budgets. Each cell shows the method difference in bold and the underlying Joint → Balanced values below it. (a) Worst task-wise projection coverage over all experts in the active collection. (b) Worst ER over the four evaluation tasks shared by all nested collections, measured on the frozen validation subsets. The outlined cell is the main capacity-constrained setting, $N = 20$ and $R = 16$.

Table 9: Random-subset robustness at $R = 16$. Confidence intervals are bootstrap intervals for the median Balanced-minus-Joint improvement in worst-task coverage.

Experts	Median ΔWorst	IQR	95% CI	Median ΔP10	Median gap reduction
4	+6.33 pp	[3.38, 8.19]	[4.76, 7.54]	+4.65 pp	10.73 pp
8	+12.82 pp	[11.57, 15.12]	[12.11, 14.38]	+6.41 pp	18.62 pp
12	+16.86 pp	[15.55, 18.29]	[16.26, 17.75]	+12.10 pp	23.08 pp

G.1 Robustness to Expert-Subset Selection

To test whether the geometric advantage depends on the nested expert sequence used in the main analysis, we sample 50 subsets each at $N \in \{4, 8, 12\}$ from the 20-expert pool and merge every subset at $R = 16$. Balanced-KnOTS improves worst-task coverage over Joint KnOTS-TIES in all 150 subsets. As summarized in Table 9, the median gain grows with the number of competing experts, consistent with the capacity-allocation interpretation.

Across all subsets, ΔWorst correlates more strongly with max–min update-energy imbalance (Spearman $\rho = 0.552$) than with $\mathrm{std}(\log \text{ energy})$ ($\rho = 0.325$) or weighted chordal diversity ($\rho = 0.080$). The within-N trends are not uniform, so these statistics identify energy imbalance as a useful pressure indicator but do not establish a monotonic dependence on subspace diversity.

G.2 Offline Construction Cost

Balanced-KnOTS adds cost only during post-hoc adapter construction. Table 10 reports median wall-clock time over three isolated runs per configuration. At 20 experts, Balanced-KnOTS requires approximately 10.5 minutes, or 5.7 times the Joint construction time. This overhead does not affect deployment: all three methods export rank-16 adapters with 43,646,976 parameters and a 166.53 MiB module-weight file.

The peak memory of Balanced-KnOTS is comparable to task-normalized construction in the measured runs. Its practical trade-off is therefore additional one-time optimization latency, not a larger deployed adapter or higher inference rank.

H Notation

Tables 11 and 12 summarize the notation used in the formulation, analysis, and method. Module indices are omitted when an expression concerns a single LoRA-modified module.

Table 10: Median shared-space construction time and Balanced-KnOTS peak resident memory.

Experts	Joint (s)	Normalized (s)	Balanced (s)	Peak RSS (MiB)
4	11.66	28.33	175.75	2818
8	33.93	43.88	270.53	4044
20	110.58	104.99	630.03	7921

Table 11: Indices, dimensions, and matrix notation.

Symbol	Definition
$i \in \{1, \ldots, N\}$	Task or expert index; N is the number of adapters being merged.
ℓ	Index of a LoRA-modified module.
d_ℓ, m_ℓ	Output and input dimensions of module ℓ.
r_i	Rank of task adapter i; the experiments use $r_i = 16$.
R	Fixed rank budget of the merged adapter.
$K = \sum_i r_i$	Total input-adapter rank for one module.
$\Delta W_{i\ell}$	Effective update learned by task i at module ℓ, including LoRA scaling.
$B_{i\ell}, A_{i\ell}$	LoRA factors satisfying $\Delta W_{i\ell} = B_{i\ell} A_{i\ell}$, with shapes $d_\ell \times r_i$ and $r_i \times m_\ell$.
U_ℓ	Orthonormal basis of the selected rank-R left subspace at module ℓ.
$\mathcal{U} = \{U_\ell\}_\ell$	Collection of shared bases across all modified modules.
Z_ℓ	Final merged coordinates, with $\Delta W_\ell^{\text{merge}} = U_\ell Z_\ell$.
$Z_{i\ell}$	Task-specific coordinates inside U_ℓ; used for projected task updates or reconstruction analysis.
L_i	Gram factor satisfying $L_i L_i^\top = A_i A_i^\top$ in exact arithmetic.
$F_i = B_i L_i$	Low-rank left factor satisfying $F_i F_i^\top = \Delta W_i \Delta W_i^\top$.
Q, Λ	Eigenvectors and eigenvalues of the covariance of projected coordinates.
$\bar{U}, \bar{Z}_i$	Canonically rotated shared basis and task coordinates used by TIES.
$\|\cdot\|_F$	Frobenius norm; I_R denotes the $R \times R$ identity matrix.

Table 12: Coverage, objective, and evaluation notation.

Symbol	Definition
$c_i(U)$	Module-level relative projection coverage of task i under basis U.
$C_i(\mathcal{U})$	Task-wise projection coverage after aggregating captured and total energy across modules.
G_C	Coverage gap, $\max_i C_i - \min_i C_i$.
$C_{\text{mean}}(U)$	Mean module-level coverage across tasks.
$C_{\text{softmin}}(U)$	Differentiable soft minimum of module-level task coverage.
$J(U)$	Balanced-KnOTS objective, $(1-\lambda)C_{\text{mean}} + \lambda C_{\text{softmin}}$.
λ	Weight assigned to the soft-min lower-tail objective.
τ	Soft-min temperature controlling emphasis on low-coverage tasks.
ϵ	Numerical stabilizer used in coverage denominators.
M_i^{merge}	Native evaluation metric of the merged model on task i.
M_i^{expert}	Native evaluation metric of the corresponding single expert.
ER_i	Expert recovery, $100 M_i^{\text{merge}} / M_i^{\text{expert}}$.
$\Delta C, \Delta\text{ER}$	Balanced-minus-Joint changes in task coverage and expert recovery.
$\Delta c_{\min,\ell}$	Balanced-minus-Joint change in worst-task module-level coverage at module ℓ.
$\mathcal{S}_N$	A fixed collection of N experts used in expert-count comparisons.