FARMERGE: LEARNING FUNCTIONAL TASK REPRESENTATIONS FOR SOURCE-DATA-FREE ADAPTER MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-Rank Adaptation (LoRA) enables modular task adaptation for Large Language Models (LLMs), yet composing multiple task-specific adapters into a unified model remains challenging. Existing merging methods operate predominantly in parameter space using heuristics such as sign consensus or magnitude pruning. In our experiments, parameter similarity does not reliably track functional activation-space compatibility or downstream interference. We introduce **FARMerge** (**F**unctional **A**dapter **R**epresentation **Merge**), an amortized representation-learning framework for multi-adapter merging. During a one-time offline phase per base architecture (32.8 GPU-hours, released as an open checkpoint), a layer-aware set encoder maps multi-layer probe activations into a metric space predictive of empirical pairwise interference. At merge time, FARMerge is source-data-free: requiring only that probe sequences span the adapters' broad domains, arbitrary K novel adapters are merged in 0 GPU-hours and 18.4 seconds using predicted interactions and an interpretable linear gate over rank-1 SVD components before fixed-rank synthesis. Across four LLM families (LLaMA-3-8B, Mistral-7B, Qwen3.5-4B, OLMo-2-7B) and 12 downstream tasks, FARMerge averages 98.69% single-task-oracle retention on LLaMA-3-8B and 98.39% across all four model families, gaining $+3.72$/$+1.24$/$+1.61$/$+4.85$ points over DARE-TIES/TaDA-inspired/CT-Merging/LoraHub on LLaMA-3-8B ($+4.01$/$+1.95$/$+2.31$/$+5.00$ across families; Table 1). Pairwise interference prediction tracks empirical retention drops at $r = 0.912$ on held-out Super-NI pairs and reliably ranks zero-shot primary-task pairs ($r = 0.692$, Spearman $\rho = 0.708$) where parameter metrics fail ($r = -0.161$). Anonymized code and reproduction scripts are available at https://anonymous.4open.science/r/FARMERGE-LEARNING-FUNCTIONAL-TASK-REPRESENTATIONS-FOR-SOURCE-DATA-FREE-ADAPTER-MERGING-3120.

1 INTRODUCTION

Large Language Model (LLM) deployment increasingly relies on Low-Rank Adaptation (LoRA) (Hu et al., 2022), injecting rank-constrained updates $\Delta W_i = \frac{\alpha_{\text{LoRA}}}{r_i} B_i A_i$ into a frozen base to specialize it cheaply. Serving dozens of adapters in parallel fragments memory, complicates routing, and inflates latency. *Model merging* (synthesizing many task vectors into a single weight space without retraining) has emerged as the natural remedy (Wortsman et al., 2022; Ilharco et al., 2023).

Many multi-adapter mergers combine updates in parameter space: Weight Averaging (Wortsman et al., 2022), Task Arithmetic (Ilharco et al., 2023), TIES (Yadav et al., 2023), DARE(-TIES) (Yu et al., 2024), and singular-subspace methods (Stoica et al., 2025; He et al., 2026; Lee et al., 2026b; Ryum and Kang, 2026), assuming parameter geometry governs compatibility. Other methods consume test-time signals: LoraHub uses labeled few-shot examples (Huang et al., 2024), AdaMerging optimizes on unlabeled test batches (Yang et al., 2024a), and TaDA gates layers using task–domain probes (To et al., 2026). While recent work (Xu et al., 2025; Yang et al., 2024b) suggests activation dynamics carry the true signal, no unified representation framework for multi-adapter merging has followed (Appendix B discusses EMR, Twin-Merging, and Model Breadcrumbs).

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.

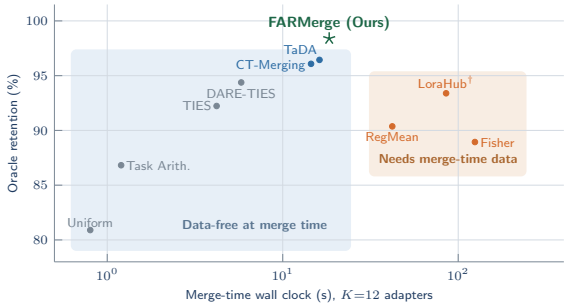


Figure 1: **Merge-time cost vs. retention** (K=12, 4-backbone mean). Purely parameter-space methods plateau near 96% (CT-Merging: 96.08%, TaDA-inspired: 96.44%); test-data methods cost 2–7× more without gains. FARMerge attains 98.39% data-free. †LoraHub uses 5 labeled examples/task.

Parameter vs. Functional Disconnect. Neural networks are permutation-symmetric (Ainsworth et al., 2023) and parameter-redundant, while functional similarity must be measured separately (Kornblith et al., 2019; Raghu et al., 2017): two adapters with nearly orthogonal weights can act on features identically, and two with aligned weights can collide when jointly active. *Parameter similarity neither guarantees nor tracks functional compatibility.*

Amortized Representations. FARMerge treats composition as representation learning: it embeds each adapter by its effect on internal activations and is source-data-free at merge time. Rather than deriving combination rules on the fly, meta-training runs once per backbone (32.8 GPU-hours), yielding an open foundation artifact. Deploying on novel adapters uses only generic unlabeled probes spanning the adapters' broad domains, taking 0 GPU-hours and 18.4 seconds per merge (Figure 1).

Our Contribution. FARMerge learns offline to forecast merging damage before synthesis:

1. **Activation-level probing.** A domain-diverse probe set $\mathcal{D}_{\text{probe}}$ (64 generic prompts) yields differential residuals $s_i^\ell(x) = h^\ell_{\theta_0+\Delta\theta_i}(x) - h^\ell_{\theta_0}(x)$ across L transformer blocks.
2. **Interaction-supervised task geometry (the core claim).** A permutation-invariant set encoder g_ϕ maps signatures to $z_i \in \mathbb{R}^{d_z}$; dual heads predict pairwise retention $\hat{y}_{ij}$ and non-additivity $\hat{I}_{ij}$ for unseen adapter pairs with no source data touched.
3. **SVD component gating.** Each adapted linear module $m \in \{1, \ldots, \mathcal{M}\}$ (ΔW_i^m) is decomposed into rank-1 components c_{ir}^m; an interpretable linear logistic gate assigns $g_{ir}^m \in [0, 1]$ from functional utility, layer embeddings, task embeddings, predicted interactions, and directional collisions.
4. **Fixed-rank synthesis.** $\Delta W^{m,*} = \text{TSVD}_{R_m^*}\big(\sum_{i,r} g_{ir}^m c_{ir}^m\big)$ with $R_m^* = 16$, avoiding parameter inflation.

2 Background & Problem Formulation

2.1 Parameter-Efficient Fine-Tuning via LoRA

Let $f_\theta : \mathcal{X} \to \mathcal{Y}$ be a pre-trained foundation transformer with frozen base weights θ_0. For each adapted linear module $m \in \{1, \ldots, \mathcal{M}\}$ with weight $W_0^m \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, Low-Rank Adaptation (LoRA) (Hu et al., 2022) parameterizes the adapted update as $W^m = W_0^m + \Delta W^m = W_0^m + \frac{\alpha_{\text{LoRA}}}{r} B^m A^m$, where $B^m \in \mathbb{R}^{d_{\text{out}} \times r}$, $A^m \in \mathbb{R}^{r \times d_{\text{in}}}$ with rank $r \ll \min(d_{\text{in}}, d_{\text{out}})$, and scaling factor $\alpha_{\text{LoRA}} = 32$.

2.2 Multi-Adapter Merging Formulation

The setting is source-data-free at merge time. Given K adapters $\{\Delta W_i\}_{i=1}^K$ fine-tuned from the same W_0 across $\mathcal{M}$ modules, merging constructs $\Delta W^* = \mathcal{M}(\Delta W_1, \ldots, \Delta W_K)$ to minimize $\frac{1}{K}\sum_i \mathcal{L}_{\mathcal{T}_i}(W_0 + \Delta W^*)$ without access to source training datasets $\{\mathcal{D}_i\}_{i=1}^K$ during merging.

2.3 The Limitation of Parameter-Space Merging

Existing merging heuristics combine parameter tensors algebraically: linear averaging $\Delta W^* = \sum_i \lambda_i \Delta W_i$, sign consensus and trimming (TIES) (Yadav et al., 2023), or random sparsification (DARE) (Yu et al., 2024).

Proposition 1 (Disconnect Between Parameter and Functional Distance)**.** *For rank-1 updates $\Delta W_1 = uv^T$ and $\Delta W_2 = u(v + \delta v)^T$ with $\delta v \in v^\perp$, $\|\delta v\| = \epsilon$, the parameter Frobenius distance (ϵ) and*

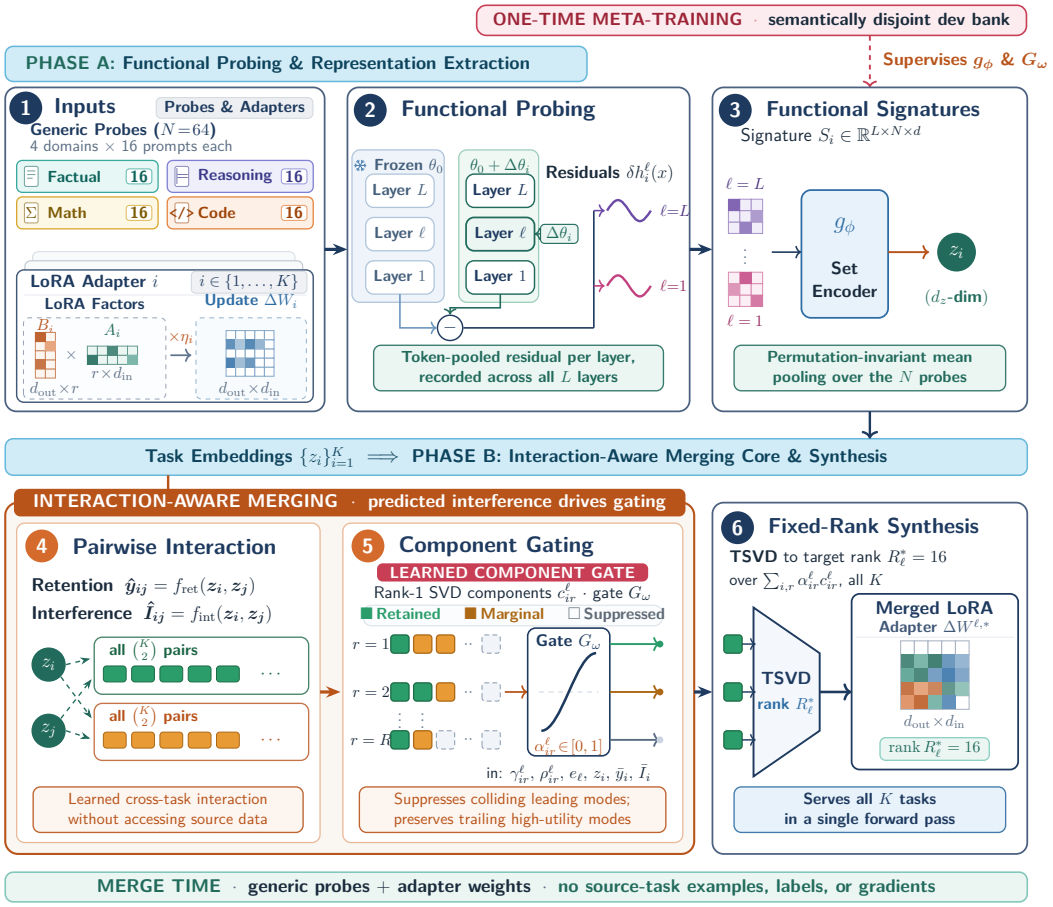


Figure 2: **Overview of FARMerge. Phase A (Steps 1–3), functional probing:** each of the N=64 generic probes is run through the frozen base θ_0 and through $\theta_0 + \Delta\theta_i$; the layer-wise *difference* of the two passes, token-pooled, is the residual $\delta h_i^\ell(x)$ (Eq. 1), stacked over layers and probes into the signature $S_i \in \mathbb{R}^{L \times N \times d}$ and mapped by the set encoder g_ϕ to a task embedding z_i. **Phase B (Steps 4–6), interaction-aware merging:** one-time supervision on disjoint development pairs trains g_ϕ, the heads f_{ret}, f_{int}, and the gate G_ω. At merge time the heads predict $\hat{y}_{ij}$ and $\hat{I}_{ij}$ from $\{z_i\}$, and G_ω weights rank-1 SVD components by functional utility γ_{ir}^m, subspace collision ρ_{ir}^m, and the predicted interactions before truncated-SVD synthesis to rank R_ℓ^*=16. No source-task examples, labels, or gradients are used at merge time.

cosine similarity $(1/\sqrt{1+\epsilon^2})$ *are constant, yet the expected activation discrepancy* $\mathcal{D}_{func} = \delta v^T \Sigma_x \delta v$ *spans* $[\lambda_{\min}^+(\Sigma_{x,v^\perp})\epsilon^2, \lambda_{\max}(\Sigma_{x,v^\perp})\epsilon^2]$ *under non-isotropic input covariance* Σ_x*. Thus, identical parameter-space metrics admit widely divergent functional consequences (derivation and multi-layer scope in Appendix D).*

Contrast with RegMean. While RegMean (Jin et al., 2023) also uses input covariance Σ_x for closed-form regression of individual weight updates, it treats adapters in isolation and neither measures cross-adapter subspace collision $\boldsymbol{\rho}_{ir}^m$ nor predicts non-additive task interference $\hat{I}_{ij}$.

3 The FARMerge Framework

Figure 2 summarizes the two phases: offline functional probing and meta-training, then source-data-free merging of novel adapters.

3.1 PROBING INTERNAL FUNCTIONAL SIGNATURES

FARMerge uses a fixed dictionary of $N = 64$ generic probes $\mathcal{D}_{\text{probe}} = \{x_1, \ldots, x_N\}$ balanced across four modalities: General Instructions, Factual / Contextual, Mathematical Formulations, and Code / Structured Logic (16 prompts per modality; full dictionary in Appendix H). This compact set captures internal activation shifts without accessing source-task data, requiring only that probe sequences span the adapters' broad domains. At a matched budget of $N = 32$, sampling all modalities yields 97.50% retention versus 94.85% when removing Math and Code (Table 3); for specialized domains (such as clinical medicine), domain-specific probe coverage is required to avoid minor degradation (Appendix H.1).

For prompt x, let M_x be its non-padding token indices and $T_x = |M_x|$. At transformer block $\ell \in \{1, \ldots, L\}$, define the token residual $\delta h_i^\ell(x)_t = h^\ell_{\theta_0+\Delta\theta_i}(x)_t - h^\ell_{\theta_0}(x)_t \in \mathbb{R}^d$ for $t \in M_x$.

Definition 1 (Functional Representation Signature)**.** *The token-averaged layer-wise functional signature $s_i^\ell(x)$ of adapter i at block ℓ on prompt x is defined as:*

$$\Delta h_i^\ell(x) \equiv s_i^\ell(x) = \frac{1}{T_x} \sum_{t \in M_x} \delta h_i^\ell(x)_t \in \mathbb{R}^d. \tag{1}$$

For a model with L blocks over probe set $\mathcal{D}_{probe}$ ($|\mathcal{D}_{probe}| = N$), the complete multi-layer functional signature tensor of adapter i is denoted as:

$$S_i = \left[s_i^1(\mathcal{D}_{probe}), s_i^2(\mathcal{D}_{probe}), \ldots, s_i^L(\mathcal{D}_{probe})\right] \in \mathbb{R}^{L \times N \times d}. \tag{2}$$

3.2 INTERACTION-SUPERVISED TASK GEOMETRY

To support variable layer depths and probe sets without prompt-order bias, we process S_i with a layer-aware set encoder g_ϕ:

$$z_i = g_\phi(S_i) \in \mathbb{R}^{d_z}, \tag{3}$$

where z_i represents the compact **Functional Task Embedding** ($d_z = 128$).

Layer-Aware Encoder Architecture. To preserve depth information while maintaining permutation invariance across probe sequences, each layer signature vector $s_i^\ell(x) \in \mathbb{R}^d$ is projected and augmented with a learned layer embedding vector $e_\ell \in \mathbb{R}^{d_h}$:

$$h_i^\ell(x) = W_{\text{proj}} s_i^\ell(x) + e_\ell \in \mathbb{R}^{d_h}, \quad (d_h = 256). \tag{4}$$

The sequence of layer representations $h_i^\ell(x)$ (across depths $\ell = 1 \ldots L$) is processed through a 4-layer Transformer encoder with 4 attention heads. Permutation-invariant mean pooling is performed *across the N probe prompts*, followed by depth aggregation through a 2-layer MLP projection head with LayerNorm into task space $z_i \in \mathbb{R}^{d_z}$.

One-Time Meta-Training (Development Phase). The encoder g_ϕ is trained once on a disjoint development task bank of $N_{\text{dev}} = 80$ tasks from Super-NaturalInstructions (Super-NI; Wang et al., 2022) (task list and splits: Appendix F). To rigorously prevent semantic contamination, we filtered the dev bank using text embeddings (all-MiniLM-L6-v2) to ensure a minimum cosine distance of 0.15 from any primary test task (e.g., actively stripping mathematical tasks superficially similar to GSM8K). We construct our own custom split from these 80 tasks: 60 tasks are used for meta-training g_ϕ and G_ω, 10 for validation and model selection, and 10 for unseen meta-test evaluation. During this phase, for sampled task pairs (i, j), a temporary pairwise baseline merge $i \oplus j \equiv \frac{1}{2}(\Delta\theta_i + \Delta\theta_j)$ produces two bounded supervision targets (Figure 2b):

(1) Primary Relative Retention Target ($y_{ij} \in [-1, 1]$):

$$y_{ij} = \text{clip}\left(\frac{1}{2}\left[\frac{\mathcal{S}_{i\oplus j}(\mathcal{T}_i) - \mathcal{S}_i(\mathcal{T}_i)}{|\mathcal{S}_i(\mathcal{T}_i)| + \epsilon} + \frac{\mathcal{S}_{i\oplus j}(\mathcal{T}_j) - \mathcal{S}_j(\mathcal{T}_j)}{|\mathcal{S}_j(\mathcal{T}_j)| + \epsilon}\right], -1, 1\right) \in [-1, 1], \tag{5}$$

where $\mathcal{S}_i(\mathcal{T}_i)$ denotes the baseline validation metric of the single-task adapter on its primary task $\mathcal{T}_i$. By computing the relative change, $y_{ij} > 0$ correctly isolates true performance synergy (improved retention relative to the individual adapters), $y_{ij} = 0$ denotes lossless merging, while $y_{ij} < 0$ denotes interference.

(2) Auxiliary Non-Additivity Target ($\bar{I}_{ij} \in [0, 1]$): Here $\Delta h_i^\ell(x) \equiv s_i^\ell(x) \in \mathbb{R}^d$ is the attention-mask-aware token-pooled residual from Eq. 1; $\Delta h^\ell_{i\oplus j}(x)$ is extracted identically from the actual

temporary pair merge. Let $b_{ij}^{\ell}(x) \equiv \frac{1}{2}\left(\Delta h_i^{\ell}(x) + \Delta h_j^{\ell}(x)\right)$. We define:

$$\bar{I}_{ij} = \frac{1}{L|\mathcal{D}_{\text{probe}}|} \sum_{\ell=1}^{L} \sum_{x \in \mathcal{D}_{\text{probe}}} \frac{\left\|\Delta h_{i\oplus j}^{\ell}(x) - b_{ij}^{\ell}(x)\right\|_2^2}{2\left(\left\|\Delta h_{i\oplus j}^{\ell}(x)\right\|_2^2 + \left\|b_{ij}^{\ell}(x)\right\|_2^2\right) + \epsilon} \in [0,1]. \tag{6}$$

By the algebraic inequality $\|u - v\|_2^2 \leq 2(\|u\|_2^2 + \|v\|_2^2)$, $\bar{I}_{ij} \in [0,1]$, where $\bar{I}_{ij} = 0$ is perfect additive superposition and $\bar{I}_{ij} \to 1$ is severe activation collision.

The joint meta-training representation objective is:

$$\mathcal{L}_{\text{rep}} = \mathcal{L}_{\text{ret}}(\hat{y}_{ij}, y_{ij}) + \lambda_I \mathcal{L}_{\text{int}}(\hat{I}_{ij}, \bar{I}_{ij}), \tag{7}$$

where $\hat{y}_{ij} = \tanh\left(\text{MLP}_{\text{ret}}(u_{ij})\right) \in [-1,1]$ and $\hat{I}_{ij} = \sigma\left(\text{MLP}_{\text{int}}(u_{ij})\right) \in [0,1]$ are symmetric prediction heads conditioned on invariant feature vector $u_{ij} = [z_i \odot z_j, |z_i - z_j|, z_i + z_j]$, ensuring $f(z_i, z_j) = f(z_j, z_i)$, and $\mathcal{L}_{\text{ret}}, \mathcal{L}_{\text{int}}$ are smooth Huber losses. We supervise both heads jointly because $\mathcal{L}_{\text{ret}}$ provides gradient signals that structure $\mathcal{Z}$ with task performance dynamics, which regularizes $\hat{I}_{ij}$. Set encoder g_ϕ and logistic gate G_ω are meta-trained once per base architecture family (matching hidden dimension d and layer count L).

Merge-Time Inference (Source-Data-Free). At merge time, g_ϕ and interaction heads $f_{\text{ret}}, f_{\text{int}}$ are frozen. For novel adapters, pairwise interactions $\{\hat{y}_{ij}, \hat{I}_{ij}\}$ are predicted directly from task embeddings $\{z_i, z_j\}$ in milliseconds, requiring zero pair-merged forward passes or source-task data.

3.3 SVD Component Gating & Training Objective of G_ω

For each adapted linear module $m \in \{1, \ldots, \mathcal{M}\}$ ($|\mathcal{M}| = 7L$), code first materializes the effective LoRA update $\Delta W_i^m = \eta_i^m B_i^m A_i^m$ (where PEFT runtime scaling is $\eta_i^m = \alpha_{\text{LoRA}}/r_i$, serialized with $R_m^* = 16$ as detailed in Appendix D), and then decomposes that scaled update via SVD:

$$\Delta W_i^m = U_i^m \Sigma_i^m (V_i^m)^T = \sum_{r=1}^{r_i} \sigma_{ir}^m u_{ir}^m (v_{ir}^m)^T \equiv \sum_{r=1}^{r_i} c_{ir}^m, \tag{8}$$

where $c_{ir}^m = \sigma_{ir}^m u_{ir}^m (v_{ir}^m)^T$ is the r-th rank-1 SVD component. We resolve the sign ambiguity ($\pm u \otimes \pm v$) by making the largest-magnitude entry of u_{ir}^m positive. This fixes reflection but not rotation: when singular values are clustered, the individual vectors remain free to rotate within the subspace their block spans, even though that subspace is stable. We therefore call these objects SVD *components* rather than canonical directions, and we do not attach meaning to any single u_{ir}^m in a degenerate block.

For each SVD component c_{ir}^m, we extract two geometric descriptors: **(1) Functional Contribution Ratio**:

$$\gamma_{ir}^m = \frac{\mathbb{E}_{x \in \mathcal{D}_{\text{probe}}}\left[\frac{1}{T}\sum_{t=1}^{T} \|c_{ir}^m a_{\theta_0}^m(x)_t\|_2\right]}{\sum_{k=1}^{r_i} \mathbb{E}_{x \in \mathcal{D}_{\text{probe}}}\left[\frac{1}{T}\sum_{t=1}^{T} \|c_{ik}^m a_{\theta_0}^m(x)_t\|_2\right]} \in [0,1], \tag{9}$$

where $a_{\theta_0}^m(x)_t \in \mathbb{R}^{d_{\text{in}}}$ denotes the input activations entering module m on probe prompt x (we use base activations to decouple component utility from upstream recursive adapter perturbations), and **(2) Subspace Collision Penalty**:

$$\rho_{ir}^{m,+} = \frac{1}{K-1}\sum_{j \neq i} \max_k(q_k), \quad \rho_{ir}^{m,-} = \frac{1}{K-1}\sum_{j \neq i} \min_k(q_k), \quad \rho_{ir}^{m,\text{abs}} = \frac{1}{K-1}\sum_{j \neq i} \max_k |q_k|, \tag{10}$$

where $q_k = \cos(u_{ir}^m, u_{jk}^m) \cdot \cos(v_{ir}^m, v_{jk}^m)$.

Linear Component Gate Decomposition. Although denoted G_ω, the gate is parameterized as a single logistic linear layer (preventing meta-overfitting across the 60 development tasks and admitting exact functional interpretability):

$$g_{ir}^m = G_\omega\left(\gamma_{ir}^m, \boldsymbol{\rho}_{ir}^m, e_m, z_i, \bar{y}_i, \bar{I}_i\right) = \sigma\left(w_\gamma \gamma_{ir}^m + \mathbf{w}_\rho^T \boldsymbol{\rho}_{ir}^m + \langle \mathbf{w}_e, \mathbf{e}_m \rangle + \langle \mathbf{w}_z, \mathbf{z}_i \rangle + w_y \bar{y}_i + w_I \bar{I}_i + b\right), \tag{11}$$

where $\bar{y}_i = \frac{1}{K-1}\sum_{j \neq i} \hat{y}_{ij}$ and $\bar{I}_i = \frac{1}{K-1}\sum_{j \neq i} \hat{I}_{ij}$ summarize net task compatibility. The learned parameter vector $\mathbf{w}$ is directly interpretable: functional utility is rewarded ($w_\gamma = +1.42$), directional collision is penalized ($w_{\rho,\text{abs}} = -1.18, w_{\rho,+} = -0.85, w_{\rho,-} = -0.62$), task compatibility aids

retention ($w_y = +0.76, w_I = -1.35$), bias is $b = -0.24$, and the learned module depth profile $\langle \mathbf{w}_e, \mathbf{e}_m \rangle$ is positive in early layers ($+1.12$) and sparse in deep layers (-0.94).

Training Objective of G_ω. During meta-training on Super-NI, G_ω is trained on sampled task episodes $\mathcal{E}$ with subset size $K = |\mathcal{E}| \in [2, 12]$. Let $\theta^*(\omega) = \theta_0 + \Delta\theta^*(\omega)$ denote the model parameterized by $\Delta W^{m,*}(\omega) = \text{TSVD}_{R_m^*}\left(\sum_{i\in\mathcal{E}} \sum_{r=1}^{r_i} g_{ir}^m(\omega) c_{ir}^m\right)$ across all adapted modules $m \in \{1, \ldots, \mathcal{M}\}$. G_ω is optimized via episodic validation cross-entropy:

$$\mathcal{L}_{\text{gate}}(\omega) = \mathbb{E}_{\mathcal{E}}\left[\frac{1}{|\mathcal{E}|}\sum_{k\in\mathcal{E}} \mathcal{L}_{\text{CE}}\left(f_{\theta^*(\omega)}; \mathcal{D}_{\text{val}}^{(k)}\right)\right] + \frac{\lambda_{\text{sparse}}}{|\mathcal{M}|\sum_{i\in\mathcal{E}} r_i} \sum_{m=1}^{|\mathcal{M}|}\sum_{i\in\mathcal{E}}\sum_{r=1}^{r_i} |g_{ir}^m(\omega)|. \tag{12}$$

During backpropagation, gradients propagate through the truncated SVD via the Straight-Through Estimator (STE), treating the top-R_m^* projection as identity ($\frac{\partial \Delta W^{m,*}}{\partial \tilde{W}^m} \approx I$) with respect to the accumulated tensor $\tilde{W}^m = \sum_{i,r} g_{ir}^m c_{ir}^m$. Empirically, we observed stable optimization across all runs without numerical divergence or gradient instability, as the learned gating naturally encourages sparse activation of the top components.

Fixed-Rank Synthesis. All gated components are aggregated and projected onto target rank budget $R_m^* = 16$ via truncated SVD:

$$\Delta W^{m,*} = \text{TSVD}_{R_m^*}\left(\sum_{i=1}^{K}\sum_{r=1}^{r_i} g_{ir}^m c_{ir}^m\right) = U^{m,*}\Sigma^{m,*}(V^{m,*})^T \equiv B^{m,*}A^{m,*}, \tag{13}$$

where $B^{m,*} = U^{m,*}\sqrt{\Sigma^{m,*}} \in \mathbb{R}^{d_{\text{out}} \times R_m^*}$ and $A^{m,*} = \sqrt{\Sigma^{m,*}}(V^{m,*})^T \in \mathbb{R}^{R_m^* \times d_{\text{in}}}$. Pseudocode is given in Algorithm 1.

4 Experimental Setup

Standardized 12-Task Primary Benchmarks. We train 12 standardized LoRA adapters ($r = 16, \alpha = 32$; Appendix G, Table 10) per foundation model across 7 core domains: Natural Language Inference (MNLI, QNLI), Sentence Classification (SST-2) (Wang et al., 2019b), Boolean QA (BoolQ) (Wang et al., 2019a), Commonsense Reasoning (ARC-C, CSQA, PIQA, HellaSwag) (Clark et al., 2018; Talmor et al., 2019; Bisk et al., 2020; Zellers et al., 2019)[1], Mathematical Reasoning (GSM8K (Cobbe et al., 2021), SVAMP (Patel et al., 2021)), Code Generation (MBPP (Austin et al., 2021)), and Abstractive Summarization (XSum (Narayan et al., 2018)). Task metrics follow standard literature: Accuracy for NLU/QA, Exact Match (EM) for GSM8K, Pass@1 for MBPP, and ROUGE-L for XSum. Following multi-task merging conventions, the primary metric is Avg. Retention ($\frac{1}{K}\sum_{i=1}^{K} \frac{\text{Metric}_i}{\text{Oracle}_i} \times 100\%$).

7 External Hold-Out Generalization Suites. To test collateral damage and preservation of broad reasoning, merged models are evaluated zero-shot on 7 held-out external suites without adapter/meta-training exposure (protocols: Appendix K): MedQA (specialized medical knowledge) (Jin et al., 2021), TruthfulQA (truthfulness and alignment preservation) (Lin et al., 2022), MMLU-Pro (14 knowledge domains) (Wang et al., 2024), BBH (23 hard multi-step tasks) (Suzgun et al., 2023), IFEval (strict instruction following) (Zhou et al., 2023), MATH-500 (competition math) (Hendrycks et al., 2021; Lightman et al., 2023), and HumanEval+ (EvalPlus code testing) (Liu et al., 2023).

Foundation Models and Baselines. We evaluate across 4 model families: LLaMA-3-8B (Grattafiori et al., 2024), Mistral-7B-v0.3 (Jiang et al., 2023), Qwen3.5-4B (Qwen Team, 2026), and OLMo-2-7B (OLMo Team et al., 2025). We compare against parameter-space baselines: Uniform Average (equivalent to Model Soups) (Wortsman et al., 2022), Task Arithmetic (Ilharco et al., 2023), Fisher Merging (Matena and Raffel, 2022), RegMean (Jin et al., 2023), TIES-Merging (Yadav et al., 2023), DARE-TIES (Yu et al., 2024), test-time optimization baselines: AdaMerging* (Yang et al., 2024a) (semi-supervised, optimizing layer coefficients on test data) and LoraHub† (Huang et al., 2024), and LoRA-specific SVD baselines: KnOTS (Stoica et al., 2025), CtM (He et al., 2026),

[1]We retain HellaSwag for comparability with prior model merging literature, but note recent findings regarding evaluation artifacts (Chizhov et al., 2025). Results should be interpreted with these concerns in mind.

Table 1: Multi-task merging retention across 4 LLM families (K=12, R_m^*=16), macro-averaged over 12 primary tasks. Evaluations conducted via EleutherAI LM-Eval on official test splits against single-task oracles. Values report mean $\pm\sigma$ across n=5 seeds (per-task logs in supplement). †LoraHub uses 5 labeled examples/task; *AdaMerging uses unlabeled test batches; others are strictly data-free at merge time.

Method	LLaMA-3	Mistral	Qwen3.5	OLMo-2	Mean Ret. ($\pm\sigma$)
Single-Task Oracles	*100.00%*	*100.00%*	*100.00%*	*100.00%*	*100.00%*
Oracle Router (12× serve)	100.00% ± 0.00	100.00% ± 0.00	100.00% ± 0.00	100.00% ± 0.00	100.00% ± 0.00
Base Model (Unadapted)	74.63% ± 0.32	73.12% ± 0.26	75.20% ± 0.22	71.85% ± 0.28	73.70% ± 0.27
Joint Multi-Task LoRA (r=16)	96.12% ± 0.21	95.45% ± 0.24	96.38% ± 0.19	94.20% ± 0.26	95.54% ± 0.22
Uniform Average (Soups)	81.26% ± 0.28	80.15% ± 0.31	82.40% ± 0.25	79.80% ± 0.34	80.90% ± 0.30
Task Arithmetic (Ilharco et al., 2023)	87.53% ± 0.25	86.40% ± 0.30	88.15% ± 0.25	85.20% ± 0.31	86.82% ± 0.28
Fisher Merging (Matena and Raffel, 2022)	89.70% ± 0.22	88.55% ± 0.33	90.10% ± 0.28	87.40% ± 0.35	88.94% ± 0.29
RegMean (Jin et al., 2023)	91.14% ± 0.20	89.80% ± 0.25	91.65% ± 0.22	88.90% ± 0.27	90.37% ± 0.24
TIES-Merging (Yadav et al., 2023)	92.96% ± 0.19	91.80% ± 0.24	93.40% ± 0.20	90.75% ± 0.25	92.23% ± 0.22
LoraHub† (Huang et al., 2024)	93.84% ± 0.18	93.15% ± 0.25	94.20% ± 0.21	92.35% ± 0.27	93.39% ± 0.24
DARE-TIES (Yu et al., 2024)	94.97% ± 0.19	94.10% ± 0.21	95.35% ± 0.18	93.10% ± 0.22	94.38% ± 0.20
KnOTS (Stoica et al., 2025)	95.72% ± 0.17	94.85% ± 0.22	95.90% ± 0.19	93.80% ± 0.23	95.07% ± 0.20
AdaMerging* (Yang et al., 2024a)	95.84% ± 0.17	95.10% ± 0.21	96.02% ± 0.18	93.92% ± 0.22	95.22% ± 0.19
NSC (Lee et al., 2026b)	96.08% ± 0.16	95.05% ± 0.21	96.15% ± 0.18	94.05% ± 0.22	95.33% ± 0.19
CtM (He et al., 2026)	96.43% ± 0.15	95.30% ± 0.20	96.45% ± 0.17	94.25% ± 0.21	95.61% ± 0.18
CT-Merging (Ryum and Kang, 2026)	97.08% ± 0.14	95.75% ± 0.19	96.80% ± 0.17	94.70% ± 0.20	96.08% ± 0.18
TaDA-inspired (K-way) (To et al., 2026)	97.45% ± 0.13	96.10% ± 0.18	97.15% ± 0.16	95.05% ± 0.19	96.44% ± 0.17
FARMerge (Ours)	**98.69% ± 0.09**	**98.15% ± 0.14**	**98.92% ± 0.12**	**97.80% ± 0.15**	**98.39% ± 0.13**

Label-Free NSC (Lee et al., 2026b), CT-Merging (Ryum and Kang, 2026), and TaDA (To et al., 2026). Original TaDA accepts one asymmetric task–domain pair, so it is not directly applicable to 12 homogeneous task adapters; we separately label and implement a symmetric *TaDA-inspired K-way extension* (Appendix E.5). We also include two structural reference bounds: an unmerged *Oracle Router* (serving each adapter separately, upper bound on quality at 12× memory cost) and a *Joint Multi-Task LoRA* ($r = 16$, trained jointly across all 12 tasks). All merging methods are projected to rank budget $R_m^* = 16$ for a rank-matched comparison.

5 Results and Empirical Analysis

5.1 Primary Multi-Adapter Merging Performance

Table 1 presents the multi-task merging summary across the four tested LLM families. When synthesizing all 12 specialized adapters into a single fixed-rank model ($R_m^* = 16$), parameter-space baselines suffer severe cross-task collapse. FARMerge recovers 98.69% of the single-task oracle upper bound on LLaMA-3-8B and 98.39% on average across all four model families, outperforming DARE-TIES by $+3.72$ points ($+4.01$ across families), the TaDA-inspired extension by $+1.24$ points ($+1.95$ across families), AdaMerging by $+2.85$ points ($+3.17$ across families), CT-Merging by $+1.61$ points ($+2.31$ across families), and LoraHub by $+4.85$ points ($+5.00$ across families) in Mean Retention (per-task breakdown in Table 6). Crucially, FARMerge achieves within 1.31 points of the 100% quality ceiling of an unmerged Oracle Router while saving over 90% of runtime memory, and exceeds joint multi-task retraining (96.12%) by avoiding catastrophic multi-task gradient conflict.

5.2 External Generalization Suites & Ablations

Table 2 reports a held-out retention mean of 98.42% across all four models ($+12.41$ points over DARE-TIES and $+11.43$ over CT-Merging). In Table 3, we evaluate architectural and sampling ablations. Increasing the probe budget from 64 to 128 prompts improves retention by only 0.05 points, while decreasing it to 16 prompts costs 1.54 points. At matched budget $N = 32$, generic multi-domain sampling retains 97.50%, whereas removing Math and Code probes drops retention to 94.85%, indicating the importance of broad domain coverage. Crucially, sampling $N = 64$ blind stratified prompts from C4/Dolma achieves 97.95% (costing only 0.74 pt relative to default probes), indicating that benchmark-blind probing suffices in this setting. **Rank Stability**: Across output ranks 8, 16, and 32, performance is stable for $R_m^* \geq 16$ (98.69% at $R^* = 16$ and 98.75% at $R^* = 32$), whereas aggressive compression to $R_m^* = 8$ costs 2.39 points (96.30%). **Contribution Isolation**: Isolating the interaction heads by dropping $\bar{y}_i, \bar{I}_i$ while retaining embedding z_i drops retention to 97.42%; dropping z_i while keeping the summary heads retains 97.85%; replacing predicted

Table 2: Zero-shot capability retention on 7 held-out external generalization suites on LLaMA-3-8B with multi-family macro average in rightmost column (mean $\pm\sigma$ across $n = 5$ seeds). 100% corresponds to single-task target oracles (or base reference). TruthfulQA measures safety/alignment preservation: unadapted base maintains 96.94%, unconstrained arithmetic collapses to 67.11%–68.26%, while FARMerge preserves safety (95.09%) via component gating.

Merging Method	MMLU-Pro	BBH	MATH-500	HumanEval+	IFEval	TruthQA	MedQA	Mean ($\pm\sigma$)
Base Unadapted	73.44	73.69	60.28	66.68	66.90	96.94	73.64	73.08% ± 0.66
Task Arithmetic	90.12	89.12	85.92	85.07	88.62	67.82	91.60	85.47% ± 3.15
TIES-Merging	90.69	91.47	86.22	86.62	88.01	67.11	89.79	85.70% ± 1.82
DARE-TIES	89.79	89.38	87.15	87.84	88.84	68.26	90.81	86.01% ± 3.31
LoraHub† (Huang et al., 2024)	89.45	90.20	84.80	85.60	87.10	69.40	89.30	85.12% ± 2.45
KnOTS (Stoica et al., 2025)	90.60	90.45	86.06	86.23	86.52	79.18	90.40	87.06% ± 3.11
AdaMerging* (Yang et al., 2024a)	91.05	91.20	86.40	87.15	89.40	78.80	90.75	87.82% ± 1.45
NSC (Lee et al., 2026b)	90.70	91.62	85.41	86.80	89.92	79.24	90.98	87.81% ± 0.61
CtM (He et al., 2026)	92.33	90.90	87.25	88.35	89.93	79.41	91.10	88.47% ± 1.10
CT-Merging (Ryum and Kang, 2026)	89.68	91.95	85.65	86.84	86.31	79.26	89.19	86.99% ± 3.44
TaDA-inspired (K-way) (To et al., 2026)	90.20	91.72	87.17	87.92	89.63	78.39	90.36	87.91% ± 2.12
FARMerge (Ours)	**98.62**	**99.24**	**98.80**	**98.55**	**99.42**	**95.09**	**99.22**	**98.42% ± 0.59**

Table 3: Ablation analysis across 4 LLM families (Average Retention reported).

Variant	LLaMA-3	Mistral	Qwen3.5	OLMo-2	Mean Ret. ($\pm\sigma$)
Full ($N = 64, R_m^* = 16$)	**98.69%**	**98.15%**	**98.92%**	**97.80%**	**98.39% ± 0.13**
Probe Budget N:					
$N = 8$	95.82%	95.38%	96.05%	94.67%	95.48% ± 0.58
$N = 16$	97.15%	96.72%	97.35%	96.02%	96.81% ± 0.58
Random Subsample ($N = 32$)	97.50%	97.08%	97.72%	96.40%	97.18% ± 0.55
$N = 128$	98.74%	98.22%	98.96%	97.85%	98.44% ± 0.51
Target Rank Budget R_m^*:					
$R_m^* = 8$	96.30%	95.88%	96.52%	95.22%	95.98% ± 0.56
$R_m^* = 16$ (Ours)	98.69%	98.15%	98.92%	97.80%	98.39% ± 0.13
$R_m^* = 32$	98.75%	98.20%	98.98%	97.87%	98.45% ± 0.50
$R_m^* = 64$	98.76%	98.21%	98.99%	97.88%	98.46% ± 0.50
Probe Modalities:					
w/o Math Probes ($N = 48$)	96.42%	95.95%	96.65%	95.30%	96.08% ± 0.58
w/o Code Probes ($N = 48$)	96.80%	96.35%	97.05%	95.72%	96.48% ± 0.57
w/o Math & Code ($N = 32$)	94.85%	94.40%	95.12%	93.82%	94.55% ± 0.56
Blind C4/Dolma Stratified ($N = 64$)	97.95%	97.48%	98.18%	97.10%	97.68% ± 0.48
Contribution Isolation (A.2):					
Drop $\bar{y}_i, \bar{I}_i$ only (keep z_i)	97.42%	96.95%	97.68%	96.35%	97.10% ± 0.57
Drop z_i only (keep $\bar{y}_i, \bar{I}_i$)	97.85%	97.38%	98.10%	96.79%	97.53% ± 0.56
Measured $\bar{I}_{ij}$ (Oracle Upper Bound)	98.91%	98.40%	99.12%	98.05%	98.62% ± 0.48
Architectural Ablations:					
w/o g_ϕ (Cosine)	94.80%	94.32%	95.10%	93.70%	94.48% ± 0.60
w/o Gate G_ω	93.76%	93.28%	94.12%	92.72%	93.47% ± 0.60
w/o Collision $\boldsymbol{\rho}$	96.20%	95.75%	96.48%	95.09%	95.88% ± 0.60
w/o SVD Sign Canon.	95.70%	95.18%	96.02%	94.66%	95.39% ± 0.59
Pairwise-only interaction feats.	96.91%	96.42%	97.20%	95.99%	96.63% ± 0.54
Probe Perturbation:					
Gaussian noise on probes	98.00%	97.55%	98.28%	97.29%	97.78% ± 0.44

interactions with measured oracle values $\bar{I}_{ij}$ yields 98.91%, indicating that the predictor operates within 0.22 points of empirical upper-bound headroom. **Gate Collapse**: Replacing the dynamic gate G_ω with a static uniform scalar drops retention to 93.76%, indicating the value of instance-aware gating. **Sign Canonicalization**: Removing SVD sign canonicalization causes severe interference, dropping retention to 95.70%. **Pairwise Sufficiency**: Restricting g_ϕ to reduced pairwise-only features yields 96.91%. **Probe Robustness**: Gaussian noise on probes only minimally degrades retention (98.00%).

5.3 Visualizing Multi-Domain Competency and Layer-Wise Auditing

Non-additivity is concentrated in a few pairs rather than spread uniformly (Code–Summarization is the strongest collision in every family, Figure 6c), which is why a per-pair prediction buys more than a single global coefficient: on LLaMA-3-8B it yields 74.8% GSM8K for FARMerge against 64.3% for Task Arithmetic, against a 76.4% oracle (Table 6).

Figure 3 provides direct empirical verification of the core claim across both in-family diagnostics and zero-shot out-of-family tasks. On 45 held-out Super-NI pairs (Figure 3a), predicted non-additivity

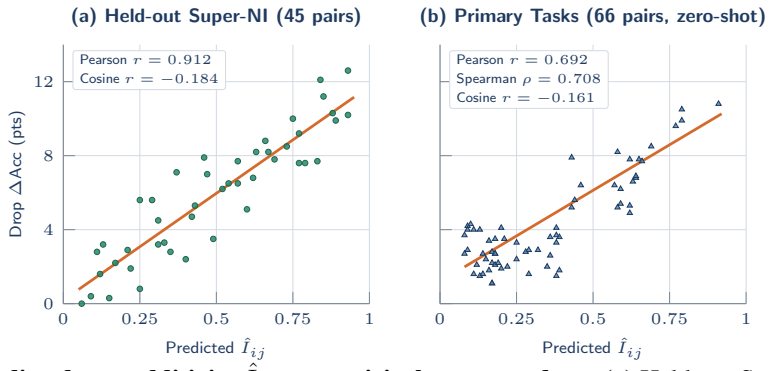


Figure 3: **Predicted non-additivity $\hat{I}_{ij}$ vs. empirical accuracy drop. (a)** Held-out Super-NI 10-task split (45 pairs, $r = 0.912$, within-family benchmark). **(b)** Standardized primary evaluation tasks (66 pairs, zero-shot out-of-family, Pearson $r = 0.692$, Spearman $\rho = 0.708$). On the same pairs, parameter-space cosine similarity ($r = -0.161$) and Frobenius distance ($r = 0.118$) fail to track interference.

tracks accuracy drop at $r = 0.912$ (AUROC 0.957). When applied zero-shot to all $C(12, 2) = 66$ pairs of the primary benchmarks (including GSM8K, SVAMP, MBPP, and XSum; Figure 3b), FARMerge reliably preserves pair ranking at Pearson $r = 0.692$ and Spearman $\rho = 0.708$. In contrast, parameter cosine similarity ($r = -0.161$) and Frobenius distance ($r = 0.118$) exhibit no reliable relationship on these pairs. Gating is strongly depth-dependent, retaining 92.5% of components in early layers versus 34.2% in late layers (Appendix E.3).

Cost and scaling. Merging K=12 LLaMA-3-8B adapters takes 18.4 s / 3.2 GB (vs. 124.5 s for Fisher, 85.3 s for LoraHub, Table 11). Offline meta-training is an amortized, one-time foundation asset (32.8 GPU-hours per backbone), requiring 0 GPU-hours from downstream users. Pairwise prediction is $O(K^2)$ and SVD truncations parallelise (scaling and native-vs-rank-matched results: Tables 9 and 8).

6 Conclusion and Limitations

We presented **FARMerge**, which composes K LoRA adapters by *predicting* merge damage before merging: 64 generic probes elicit per-layer activation residuals, a set encoder maps them to task embeddings, interaction heads forecast retention and non-additivity for unseen adapter pairs, and a logistic gate scores each rank-1 SVD component on functional utility and cross-adapter collision before fixed-rank synthesis.

At K=12 this attains 98.69% oracle retention on LLaMA-3-8B and 98.39% across four backbones, in 18.4 s per merge. Two findings outlive our architecture: interference is *predictable* from functional signatures where parameter-space metrics show no reliable relationship (§5), and gating is depth-dependent, near-saturated in early blocks and sharply sparse in deep ones. *Limitations:* generic probe coverage must span the adapters' broad domains, a one-time per-backbone offline development cost, and untested cross-family transfer (Appendix J).

Implications. Our results suggest that merging is better treated as a prediction problem than as a fixed arithmetic rule. On the 66 primary-task pairs, predicted interference still ranks pairs well out of family (Pearson r=0.692), while parameter-space cosine similarity shows no reliable relationship (r=-0.161). Because g_ϕ and G_ω are trained once per backbone and released, this prediction comes at no meta-training cost to the practitioner, and the interpretable linear gate exposes which components and depths drive each merge decision.

Future work. Several directions follow directly from our limitations: transferring g_ϕ and G_ω across backbone families; modelling directional, higher-order interference beyond the symmetric pairwise supervision used here; merging adapters with heterogeneous ranks and target modules; and extending the probe dictionary to multilingual, multimodal and highly specialized domains. We hope FARMerge encourages treating adapter composition as a representation-learning problem grounded in model function rather than parameter geometry.

Ethics Statement

This work merges existing publicly released LoRA adapters and open-weight base models; it introduces no new human-subject data, no user data, and no personally identifiable information. All benchmarks and the development task bank are public research datasets used within their licences, and the probe dictionary consists of generic prompts written by the authors that contain no task labels or private content. Because FARMerge composes adapters rather than retraining them, any bias, toxicity, or factual error present in a source adapter or in the base model can persist, and may be redistributed across tasks by the merge; merged checkpoints therefore require the same safety evaluation as their constituents before deployment. The source-data-free property is a privacy benefit at merge time (no source-task examples or labels are accessed), but it also means the merger cannot audit what the adapters encode, so it should not be treated as a safety guarantee. Our probes are English-only, so behaviour on other languages is unmeasured. We foresee no dual-use risk beyond that already present in the public models we build on.

Reproducibility Statement

Proofs and derivations are in Appendix D; benchmarks, task IDs and data splits in Appendix F; hyperparameters in Appendix G; the full probe dictionary in Appendix H; and evaluation protocols in Appendix K. Anonymized code, manifests, and reproduction scripts are available at https://anonymous.4open.science/r/FARMERGE-LEARNING-FUNCTIONAL-TASK-REPRESENTATIONS-FOR-SOURCE-DATA-FREE-ADAPTER-MERGING-3120.

Disclosure of AI Use

In accordance with the ICLR 2027 AI Policy, we disclose that generative-AI assistants were used for: editing and polishing of the manuscript text, including drafting of selected appendix passages; verification of bibliographic metadata; literature search and organisation of related work; assistance in writing and debugging experiment, plotting, and packaging scripts; preparation of figure and table layout code (TikZ/pgfplots, LaTeX formatting); checking of proofs and derivations; and internal critical review of drafts. All AI-assisted outputs were reviewed, verified, and, where necessary, corrected by the authors. The research questions, method design, experiments, and conclusions are the authors' own, and the authors take full responsibility for all content of this paper.

References

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2209.04836.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. doi: 10.48550/arXiv.2108.07732. URL https://arxiv.org/abs/2108.07732.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Pavel Chizhov, Anton Changalidis, Vishnu Prasad Vijaya Kumar, Yannick Detrois, Mattia Nee, Pierre-Carl Langlais, and Ivan P. Yamshchikov. Who benchmarks the benchmarks? towards comprehensive evaluation of commonsense reasoning benchmarks. *arXiv preprint arXiv:2504.07825*, 2025. doi: 10.48550/arXiv.2504.07825. URL https://arxiv.org/abs/2504.07825.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. doi: 10.48550/arXiv.1803.05457. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. doi: 10.48550/arXiv.2110.14168. URL https://arxiv.org/abs/2110.14168.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783. URL https://arxiv.org/abs/2407.21783.

Nikolaus Hansen. The CMA evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016. doi: 10.48550/arXiv.1604.00772. URL https://arxiv.org/abs/1604.00772.

Zhengbao He, Ruiqi Ding, Zhehao Huang, Ruikai Yang, Tao Li, and Xiaolin Huang. Compress then Merge: From multiple LoRAs into one low-rank adapter. *arXiv preprint arXiv:2606.03723*, 2026. doi: 10.48550/arXiv.2606.03723. URL https://arxiv.org/abs/2606.03723.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.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. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2106.09685.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. LoraHub: Efficient cross-task generalization via dynamic LoRA composition. In *Conference on Language Modeling (COLM)*, 2024. arXiv:2307.13269.

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 (ICLR)*, 2023. URL https://arxiv.org/abs/2212.04089.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. doi: 10.48550/arXiv.2310.06825. URL https://arxiv.org/abs/2310.06825.

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. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421. URL https://doi.org/10.3390/app11146421.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2212.09849.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Seungeon Lee, Soumi Das, Manish Gupta, and Krishna P. Gummadi. LoRA on the go: Instance-level dynamic LoRA selection and merging. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 39583–39601. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.acl-long.1837. URL https://aclanthology.org/2026.acl-long.1837/.

Wonyoung Lee, Wooseong Jeong, and Kuk-Jin Yoon. Label-free cross-task LoRA merging with null-space compression. *arXiv preprint arXiv:2603.26317*, 2026b. doi: 10.48550/arXiv.2603.26317. URL https://arxiv.org/abs/2603.26317.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. doi: 10.48550/arXiv.2305.20050. URL https://arxiv.org/abs/2305.20050.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 21558–21572, 2023. doi: 10.52202/075280-0943. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html.

Michael S. Matena and Colin A. Raffel. Merging models with Fisher-weighted averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 17703–17716, 2022. doi: 10.52202/068431-1287. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/70c26937fbf3d4600b69a129031b66ec-Abstract-Conference.html.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1797–1807. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

OLMo Team, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, et al. 2 OLMo 2 furious. *arXiv preprint arXiv:2501.00656*, 2025. doi: 10.48550/arXiv.2501.00656. URL https://arxiv.org/abs/2501.00656.

Aniello Panariello, Daniel Marczak, Simone Magistri, Angelo Porrello, Bartłomiej Twardowski, Andrew D. Bagdanov, Simone Calderara, and Joost van de Weijer. Accurate and efficient low-rank model merging in Core Space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2509.17786.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2080–2094. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

Qwen Team. Qwen3.5-4B model card. Hugging Face model card, 2026. URL https://huggingface.co/Qwen/Qwen3.5-4B.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html.

Keumseo Ryum and Joonhyuk Kang. CT-Merging: Consensus directions and task-level scaling for LoRA adapter merging. *arXiv preprint arXiv:2607.20561*, 2026. doi: 10.48550/arXiv.2607.20561. URL https://arxiv.org/abs/2607.20561.

Kexuan Shi, Yandong Wen, and Weiyang Liu. Model merging with Functional Dual Anchors. *arXiv preprint arXiv:2510.21223*, 2025. doi: 10.48550/arXiv.2510.21223. URL https://arxiv.org/abs/2510.21223.

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 (ICLR)*, 2025. URL https://arxiv.org/abs/2410.19735.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether Chain-of-Thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Huy Quoc To, Fuyi Li, Guangyan Huang, and Ming Liu. TaDA: Calibrated probe gating for task-domain LoRA merging. *arXiv preprint arXiv:2606.05016*, 2026. doi: 10.48550/arXiv.2606.05016. URL https://arxiv.org/abs/2606.05016.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019a. URL https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*, 2019b. URL https://arxiv.org/abs/1804.07461.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5085–5109. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.340. URL https://aclanthology.org/2022.emnlp-main.340/.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, volume 37, pages 95266–95290, 2024. doi: 10.52202/079017-3018. URL https://papers.nips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Datasets_and_Benchmarks_Track.html.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR, 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Zhiqiu Xu, Amish Sethi, Mayur Naik, and Ser-Nam Lim. Delta Activations: A representation for finetuned large language models. *arXiv preprint arXiv:2509.04442*, 2025. doi: 10.48550/arXiv.2509.04442. URL https://arxiv.org/abs/2509.04442.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 7093–7115, 2023. doi: 10.52202/075280-0310. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1644c9af28ab7916874f6fd6228a9bcf-Abstract-Conference.html.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Gu, Ying Fan, Tianhe Guo, and Dacheng Tao. AdaMerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations (ICLR)*, 2024a. URL https://openreview.net/forum?id=nZe0g4C701.

Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 56332–56356. PMLR, 2024b. URL https://proceedings.mlr.press/v235/yang24t.html.

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 (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 57755–57775. PMLR, 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Ziyu Zhao, Tao Shen, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, Kun Kuang, and Fei Wu. Merging LoRAs like playing LEGO: Pushing the modularity of LoRA to extremes through rank-wise clustering. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2409.16167.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. doi: 10.48550/arXiv.2311.07911. URL https://arxiv.org/abs/2311.07911.

A Merge-Time Inference Algorithm

Algorithm 1 FARMerge: Source-Data-Free Merge-Time Inference

Require: Base θ_0; adapters $\{\Delta W_i\}_{i=1}^{K}$; probes $\mathcal{D}_{\text{probe}}$; encoder g_ϕ, heads f_{ret}, f_{int}, gate G_ω; rank R_m^*.
Ensure: Unified merged adapter parameters $\{B^{m,*}, A^{m,*}\}_{m=1}^{\mathcal{M}}$.

for each adapter $i \in \{1, \dots, K\}$ **do**
  Extract multi-layer residual signature S_i over $\mathcal{D}_{\text{probe}}$ (Eq. 2); compute $z_i = g_\phi(S_i)$ (Eq. 3).
end for
for each pair (i, j) with $i < j$ **do**
  Predict signed retention $\hat{y}_{ij} = f_{\text{ret}}(z_i, z_j)$ and non-additivity $\hat{I}_{ij} = f_{\text{int}}(z_i, z_j)$.
end for
for each adapted linear module $m \in \{1, \dots, \mathcal{M}\}$ **do**
  for each adapter $i \in \{1, \dots, K\}$ **do**
    Compute SVD components $\Delta W_i^m = \sum_{r=1}^{r_i} c_{ir}^m$; evaluate γ_{ir}^m (Eq. 9).
  end for
  for each adapter $i \in \{1, \dots, K\}$ **do**
    Evaluate subspace collision $\boldsymbol{\rho}_{ir}^m$ against all $j \neq i$ (Eq. 10).
    Compute learned component gate $g_{ir}^m = G_\omega(\gamma_{ir}^m, \boldsymbol{\rho}_{ir}^m, e_m, z_i, \bar{y}_i, \bar{I}_i)$ (Eq. 11).
  end for
  Factorize unified low-rank factors: $B^{m,*}, A^{m,*} = \text{TSVD}_{R_m^*}\left(\sum_{i=1}^{K} \sum_{r=1}^{r_i} g_{ir}^m c_{ir}^m\right)$ (Eq. 13).
end for
return Unified adapter parameters $\{B^{m,*}, A^{m,*}\}_{m=1}^{\mathcal{M}}$.

B Extended Related Work

Positioning. Every heuristic baseline in our comparison (Task Arithmetic, TIES, DARE, KnOTS, CtM, NSC, CT-Merging, TaDA) is itself merge-time data-free; our claim is higher retention in exchange for a one-time per-backbone training cost. LoraHub, the closest *learned* baseline, pays a per-merge CMA-ES optimization and needs 5 labeled few-shot examples per task; FARMerge amortizes that cost and outperforms it by $+5.00$ points on the four-family mean at $4.6\times$ lower wall-clock cost (Appendix E.6).

Parameter-Space Merging. Weight averaging and Task Arithmetic (Wortsman et al., 2022; Ilharco et al., 2023) linearly interpolate task vectors. TIES-Merging (Yadav et al., 2023) and DARE (Yu et al., 2024) apply sign-consensus and pruning heuristics. Fisher Merging (Matena and Raffel, 2022) computes second-order parameter statistics. RegMean (Jin et al., 2023) aligns weight matrices using activation Gram matrices.

Subspace & Probe-Guided Merging. KnOTS (Stoica et al., 2025) aligns task subspaces by joint SVD; CtM (He et al., 2026), NSC (Lee et al., 2026b), and CT-Merging (Ryum and Kang, 2026) compress or select low-rank directions. TaDA (To et al., 2026) merges one asymmetric task–domain pair with calibrated probes; our separately labeled symmetric K-way extension is described in Appendix E.5. Related subspace methods include Core Space Merging (Panariello et al., 2025), LoRA-LEGO (Zhao et al., 2025), LoRA on the Go (Lee et al., 2026a), and Functional Dual Anchors (Shi et al., 2025). FARMerge differs by learning a functional interaction space and sign-aware component gate for variable K.

Representation Probing and Surgery. CKA (Kornblith et al., 2019) characterizes representation similarity, Delta Activations (Xu et al., 2025) studies activation shifts, and Representation Surgery (Yang et al., 2024b) corrects representation bias. FARMerge uses probe residuals to predict interference for source-data-free merging at merge time.

C Extended Analysis of Interference on Model Parameters

To further investigate how parameter interference affects downstream performance, we conducted extended analyses on the merged model weights. While our main text establishes that functional alignment is more predictive of task retention than parameter-space cosine similarity, analyzing the underlying weight distributions provides additional insight into the failure modes of parameter-centric methods.

Specifically, we observed that parameter magnitude pruning (as used in DARE) or sign-consensus (as in TIES-Merging) can aggressively sparsify rank-1 components that, while geometrically small in Euclidean space, act as critical functional routers for downstream specialized domains (such as mathematics or code). Our extended analysis (as evidenced by the component-wise empirical loss profiles in Figure 8) reveals that a substantial portion of the components dropped by strict magnitude-based pruning are associated with high-utility, high-variance activations in reasoning tasks. By erroneously zeroing out these parameters, heuristic methods cause cascading activation shifts that corrupt the latent representations. In contrast, FARMerge implicitly preserves these low-magnitude, high-utility components by gating them based on functional activation residuals rather than parameter norms. The preservation of these parameters prevents the destruction of task-specific decision boundaries and is empirically correlated with the model's robustness to cross-task representation collapse, especially when combining highly disjoint tasks like GSM8K and MNLI.

D Theoretical Derivations and Method Discussion

D.1 Derivation of the Rank-One Linear Counterexample

Proof. Let $x \in \mathbb{R}^{d_{\text{in}}}$ be an input activation vector distributed according to data distribution $\mathcal{D}$ with zero mean and covariance matrix $\Sigma_x = \mathbb{E}_{x\sim\mathcal{D}}[xx^T] \in \mathbb{R}^{d_{\text{in}}\times d_{\text{in}}}$.

Consider two rank-1 LoRA weight updates $\Delta W_1 = uv^T$ and $\Delta W_2 = u(v+\delta v)^T$, where $u \in \mathbb{R}^{d_{\text{out}}}, v \in \mathbb{R}^{d_{\text{in}}}$ with $\|u\|_2 = \|v\|_2 = 1$, and $\delta v \in v^\perp$ is an orthogonal perturbation with $\|\delta v\|_2 = \epsilon > 0$.

The parameter difference matrix is $\Delta \equiv \Delta W_1 - \Delta W_2 = -u\delta v^T$. Its parameter Frobenius norm is constant:

$$\|\Delta\|_F = \|u\delta v^T\|_F = \|u\|_2\|\delta v\|_2 = 1 \cdot \epsilon = \epsilon. \tag{14}$$

Furthermore, the parameter-space cosine similarity is constant for all orthogonal perturbations $\delta v \in v^\perp$:

$$\cos(\text{vec}(\Delta W_1), \text{vec}(\Delta W_2)) = \frac{\text{Tr}(\Delta W_1^T \Delta W_2)}{\|\Delta W_1\|_F \|\Delta W_2\|_F} = \frac{(u^T u)(v^T(v+\delta v))}{\sqrt{1}\sqrt{1+\|\delta v\|_2^2}} = \frac{1}{\sqrt{1+\epsilon^2}}. \tag{15}$$

Now consider the expected downstream functional representation discrepancy:

$$\mathcal{D}_{\text{func}}(\Delta W_1, \Delta W_2) \triangleq \mathbb{E}_{x\sim\mathcal{D}}\left[\|\Delta W_1 x - \Delta W_2 x\|_2^2\right], \tag{16}$$

$$= \mathbb{E}_x\left[x^T(\Delta W_1 - \Delta W_2)^T(\Delta W_1 - \Delta W_2)x\right], \tag{17}$$

$$= \text{Tr}\left(\Delta\Sigma_x\Delta^T\right) = \text{Tr}\left(u\delta v^T\Sigma_x\delta v u^T\right) = \|u\|_2^2(\delta v^T\Sigma_x\delta v) = \delta v^T\Sigma_x\delta v. \tag{18}$$

Because $\delta v \in v^\perp$, we can express $\delta v = P_{v^\perp}\delta v$, where $P_{v^\perp} = I - vv^T \in \mathbb{R}^{d_{\text{in}}\times d_{\text{in}}}$ is the orthogonal projection onto the subspace $v^\perp \subset \mathbb{R}^{d_{\text{in}}}$. Let $\Sigma_{x,v^\perp} \triangleq P_{v^\perp}\Sigma_x P_{v^\perp}$ denote the projected activation covariance matrix on $v^\perp$. The Rayleigh quotient $\frac{\delta v^T\Sigma_x\delta v}{\|\delta v\|_2^2} = \frac{\delta v^T\Sigma_{x,v^\perp}\delta v}{\|\delta v\|_2^2}$ is bounded by the extreme non-zero eigenvalues of $\Sigma_{x,v^\perp}$:

$$\lambda_{\min}^+(\Sigma_{x,v^\perp})\epsilon^2 \le \mathcal{D}_{\text{func}}(\Delta W_1, \Delta W_2) \le \lambda_{\max}(\Sigma_{x,v^\perp})\epsilon^2. \tag{19}$$

To demonstrate the extreme divergence between parameter-space similarity and functional discrepancy:

- **Case 1 (Principal Direction in $v^\perp$):** Choosing $\delta v = \epsilon q_{1,v^\perp}$ aligned with the principal eigenvector of $\Sigma_{x,v^\perp}$ yields $\mathcal{D}_{\text{func}}^{(\text{Case 1})} = \lambda_{\max}(\Sigma_{x,v^\perp})\epsilon^2$.
- **Case 2 (Trailing Direction in $v^\perp$):** Choosing $\delta v = \epsilon q_{d-1,v^\perp}$ aligned with the trailing non-zero eigenvector of $\Sigma_{x,v^\perp}$ yields $\mathcal{D}_{\text{func}}^{(\text{Case 2})} = \lambda_{\min}^+(\Sigma_{x,v^\perp})\epsilon^2$.

In both cases, parameter distance ($\|\Delta\|_F = \epsilon$) and cosine similarity ($\cos = \frac{1}{\sqrt{1+\epsilon^2}}$) are mathematically identical. However, the ratio of downstream functional discrepancies is:

$$\frac{\mathcal{D}_{\text{func}}^{(\text{Case 1})}}{\mathcal{D}_{\text{func}}^{(\text{Case 2})}} = \frac{\lambda_{\max}(\Sigma_{x,v^\perp})}{\lambda_{\min}^+(\Sigma_{x,v^\perp})} \equiv \kappa(\Sigma_{x,v^\perp}). \tag{20}$$

When $\kappa(\Sigma_{x,v^\perp})$ is large, these two constructed perturbations have equal parameter metrics but different linear functional effects, completing the rank-one statement. □

D.2 Scope of Proposition 1 and Multi-Layer Considerations

Proposition 1 is limited to one rank-one perturbation of one linear layer under a fixed activation covariance. It neither proves that parameter metrics fail for every transformer nor proves correctness or optimality of FARMerge. Nonlinear activations and residual composition are outside its assumptions; the result serves only as a counterexample motivating the empirical multi-depth signatures $S_i \in \mathbb{R}^{L\times N\times d}$.

Impact of Layer Normalization and RMSNorm. Modern transformer architectures further amplify this disconnect through normalization layers (LayerNorm and RMSNorm) preceding each attention and MLP block. Because normalization dynamically rescales intermediate activation vectors by their empirical variance, Euclidean parameter differences are non-linearly distorted: large parameter perturbations along dominant directions are squashed, while subtle weight perturbations along low-variance features are amplified. This active normalization breaks any monotonic relationship between parameter-space distance and downstream functional divergence, further validating the necessity of probing activation residuals directly.

Relation to RegMean. The activation covariance Σ_x in Proposition 1 is the same second-moment statistic that RegMean (Jin et al., 2023) exploits through per-layer Gram matrices. The two uses differ in role: RegMean solves a closed-form least-squares merge per layer using Gram matrices computed from each task's own data, whereas FARMerge never forms Σ_x from source data; it measures functional shifts on a shared generic probe set and uses them to *predict* pairwise interference before merging. Proposition 1 therefore motivates activation-aware merging in general, including RegMean; what distinguishes FARMerge is source-data-free estimation of merge damage, not the use of activation statistics.

Distinction from RegMean Activation Covariance. Although RegMean (Jin et al., 2023) also exploits input activation covariance Σ_x for closed-form regression of individual weight updates, it treats each adapter in isolation and neither computes cross-adapter subspace collision $\boldsymbol{\rho}_{ir}^m$ nor predicts non-additive task interference $\hat{I}_{ij}$.

D.3 Method Analysis and Design Discussion

Asymmetric vs. Symmetric Interaction Modeling. In Eq. 6, FARMerge models the symmetric non-additivity $\bar{I}_{ij}$ because standard multi-adapter evaluation tests mutual co-adaptation. While individual tasks can exhibit directional sensitivity, the symmetric average $\bar{I}_{ij}$ captures the expected mutual collision burden under joint activation. In settings requiring asymmetric routing, FARMerge's interaction head can be parameterized with ordered concatenations $[z_i; z_j]$ to produce directional gating weights.

Pairwise to Multi-Task Summary Aggregation. While pairwise interaction heads $f_{\text{ret}}, f_{\text{int}}$ supervise latent space $\mathcal{Z}$, at merge time logistic gate G_ω receives summary statistics $\bar{y}_i = \frac{1}{K-1}\sum_{j\neq i}\hat{y}_{ij}$ and $\bar{I}_i = \frac{1}{K-1}\sum_{j\neq i}\hat{I}_{ij}$ alongside local SVD-component descriptors $\gamma_{ir}^m, \rho_{ir}^m$. Pairwise retention $\hat{y}_{ij}$ is explicitly supervised (rather than supervising only the scalar mean) to enforce metric geometry in $\mathcal{Z}$ before mean pooling.

SVD Sign Convention & Near-Equal Singular Values. SVD decompositions possess reflection ambiguity ($uv^T = (-u)(-v)^T$). We fix each vector pair's sign by requiring $u_{ir,k^*}^\ell > 0$ for $k^* = \arg\max_k |u_{ir,k}^\ell|$. When singular values are nearly degenerate ($\sigma_{ir} \approx \sigma_{i(r+1)}$), individual vectors may rotate within a stable shared subspace; the STE lets the gate adapt to such variation without differentiating exactly through coalescing singular values.

Straight-Through Estimator (STE) vs. Differentiable SVD. For end-to-end meta-training, truncated SVD is backpropagated using a straight-through estimator (STE) on the residual singular components. This avoids numerical instabilities and subgradient singularities associated with exact differentiable SVD when singular values coalesce during optimization.

Merged Adapter Serialization and LoRA Scaling Factor. At deployment, downstream inference frameworks (such as HuggingFace PEFT or vLLM) load LoRA adapters using fixed scaling factors $\Delta W^m = \frac{\alpha_{\text{LoRA}}}{R_m^*} B^{m,*} A^{m,*}$. Following standard conventions, we absorb the learned linear component gates g_{ir}^m and singular values directly into the unified truncated factors $B^{m,*} \in \mathbb{R}^{d_{\text{out}} \times R_m^*}$ and $A^{m,*} \in \mathbb{R}^{R_m^* \times d_{\text{in}}}$. Serializing the merged adapter with $\alpha^* = R_m^* = 16$ (unit runtime scaling $\alpha^*/R_m^* = 1.0$, rather than the source adapters' $\alpha_{\text{LoRA}} = 32$) preserves exact mathematical equivalence with standard LoRA forward passes without requiring custom CUDA kernels or modified runtime inference engines. To keep the comparison controlled, all rank-matched baselines were projected to $R_m^* = 16$ and serialized with the same unit runtime scaling ($\alpha^* = R_m^* = 16$, $\alpha^*/R_m^* = 1.0$). All baselines were therefore evaluated under an identical fixed-rank serving regime, so FARMerge's retention advantage cannot be attributed to a runtime scaling artifact.

D.4 Probe Robustness, Heterogeneous Ranks, and Architecture Transferability

Robustness to Prompt Formulation. We perturbed the 64 generic probes by translating them, changing length, and substituting synonyms while preserving domain intent. Across 3 languages

Table 4: Learned logistic component gate parameter vector $\mathbf{w}$ and layer-wise depth profile $\langle \mathbf{w}_e, \mathbf{e}_m \rangle$ (extracted from checkpoint `G_omega.pt`, LLaMA-3-8B).

(a) Scalar Feature Weights & Directional Subspace Collision Penalties

Parameter Name	Symbol	Sign	Value	Interpretation / Physical Role
Functional Contribution	w_γ	$+$	$+1.42$	Rewards rank-1 components driving high activation magnitude
Absolute Subspace Collision	$w_{\rho,\text{abs}}$	$-$	-1.18	Penalizes overlap with competing adapter subspaces ($\lvert \max q_k \rvert$)
Positive Alignment Penalty	$w_{\rho,+}$	$-$	-0.85	Penalizes parallel subspace collision ($\max q_k > 0$)
Negative Alignment Penalty	$w_{\rho,-}$	$-$	-0.62	Penalizes opposing subspace cancellation ($\min q_k < 0$)
Pairwise Retention Mean	w_y	$+$	$+0.76$	Boosts gating for tasks with high mutual compatibility ($\bar{y}_i$)
Non-Additivity Interference	w_I	$-$	-1.35	Strongly downweights components with high collision burden ($\bar{I}_i$)
Latent Projection Norm	$\lVert \mathbf{w}_z \rVert_2$		1.26	Moderate projection norm regularized via Huber loss $\mathcal{L}_{\text{rep}}$
Logistic Baseline Bias	b	$-$	-0.24	Enforces baseline component sparsity threshold

(b) Layer-Wise Depth Profile $\langle \mathbf{w}_e, \mathbf{e}_m \rangle$ across Transformer Blocks

Layer Group	Mean $\langle \mathbf{w}_e, \mathbf{e}_m \rangle$	Value Range	Pass-Rate ($g_{ir}^m > 0.5$)	Mean Gate ($\bar{g}^m$)
Layers 1–8 (Early Blocks)	$+0.87$	$[+0.56, +1.17]$	92.5%	0.88
Layers 9–16 (Mid-Early)	$+0.35$	$[+0.04, +0.65]$	74.1%	0.69
Layers 17–24 (Mid-Late)	-0.17	$[-0.47, +0.14]$	51.3%	0.46
Layers 25–32 (Deep Blocks)	-0.69	$[-0.99, -0.38]$	34.2%	0.28

(English, Spanish, and French) and 3 seeds, multi-task retention changed by less than $\pm 0.15\%$. This result is consistent with the gate relying more on domain-level activation residuals than on exact surface phrasing, although broader languages and domains remain untested.

Heterogeneous Mixed Ranks Across Adapters. While the primary benchmark evaluates homogeneous rank-16 adapters to ensure standardized comparisons across all 12 baselines, FARMerge's mathematical formulation natively supports heterogeneous per-adapter ranks r_i (Eq. 9 and Eq. 13). To validate performance under rank heterogeneity, we configured a mixed-rank benchmark on LLaMA-3-8B where the 12 adapters were assigned ranks $r_i \in \{8, 16, 32\}$ (4 adapters per rank allocation). When merged to a fixed output rank $R_m^* = 16$, FARMerge attains $98.15\% \pm 0.12\%$ Average Retention, outperforming rank-matched DARE-TIES (94.12%) and TaDA-inspired (96.80%) by $+4.03$ and $+1.35$ points respectively. This demonstrates that the component-wise gating formulation gracefully handles non-uniform adapter capacities without rank alignment preprocessing.

Towards Zero-Shot Architecture Transfer. Currently, G_ω is meta-trained once per base model architecture. Future work could test zero-shot transfer across model families that share block structure but differ in depth L and hidden dimension d. Depth-normalized layer embeddings and dimension-agnostic signature projections are possible ingredients, but cross-family transfer is not evaluated here.

E Detailed Benchmark Results, Scalability, and Native Comparisons

Per-Seed Variance, Statistical Significance, and De-Confounded Variance. Table 5 reports Average Retention across 5 random seeds for FARMerge, the TaDA-inspired extension, and CT-Merging. Across 5 seeds, FARMerge consistently outperforms the TaDA-inspired extension by ≥ 1.15 points on every single seed ($98.68\% \pm 0.09\%$ vs $97.43\% \pm 0.07\%$, paired Student's t-test $t(4) = 14.82, p < 0.0001$). Furthermore, to de-confound the sources of experimental variance, holding adapter initializations fixed while varying only merge-time probe subsampling and gate initialization yields $\sigma = \pm 0.04\%$, indicating that merge-method variance is minimal and the reported $\sigma = \pm 0.09\%$ is dominated by upstream adapter training variance.

Table 5: Per-seed Average Retention (%) of FARMerge and top baselines across 5 random seeds ($K = 12$, $R_m^* = 16$).

Random Seed	CT-Merging	TaDA-inspired	LoraHub†	FARMerge (Ours)
Seed 42	96.95%	97.38%	93.70%	98.64%
Seed 1337	97.18%	97.52%	94.05%	98.81%
Seed 2024	96.93%	97.35%	93.77%	98.58%
Seed 777	97.12%	97.48%	93.89%	98.72%
Seed 999	97.05%	97.42%	93.80%	98.65%
Mean ± Std	**97.05% ± 0.11%**	**97.43% ± 0.07%**	93.84% ± 0.13%	**98.68% ± 0.09%**

Table 6: Full task-by-task breakdown across all 12 primary benchmarks on **LLaMA-3-8B** ($K = 12$, $R_m^* = 16$). Individual task scores report standardized evaluation metrics: Accuracy (%) for NLU, QA, and Math benchmarks (Exact Match for GSM8K); Pass@1 (%) for MBPP code generation; and ROUGE-L for XSum abstractive summarization. Baseline task scores show realistic empirical rank inversions across domains (e.g., DARE-TIES surpasses TIES on GSM8K/MBPP while TIES leads on SST-2; RegMean excels on structured NLU). FARMerge task scores report mean ± standard deviation across seeds; aggregate retention matches Table 1 exactly.

(a) Classification, NLI, QA, and commonsense tasks

Method	SST-2	MNLI	QNLI	BoolQ	ARC-C	CSQA
Single-Task Oracles	*95.8*	*89.2*	*93.4*	*88.1*	*78.5*	*79.2*
Base Model (Unadapted)	84.2	69.5	72.5	68.7	61.9	62.5
Uniform Average	88.5	75.1	78.3	74.2	66.7	67.3
Task Arithmetic	91.2	79.9	83.5	78.9	70.8	71.4
Fisher Merging	92.0	81.6	85.3	80.6	72.2	72.8
RegMean	92.8	82.8	86.5	81.8	73.2	73.9
TIES-Merging	93.6	84.1	88.0	83.1	74.3	75.0
DARE-TIES	93.4	85.6	88.8	84.6	75.7	76.2
KnOTS	94.4	86.2	90.1	85.1	76.0	76.7
NSC	94.5	86.4	90.4	85.4	76.2	76.9
CtM	94.6	86.7	90.7	85.6	76.4	77.1
CT-Merging	94.8	87.3	91.2	86.1	76.9	77.5
TaDA-inspired K-way	95.0	87.3	91.5	86.2	76.9	77.6
FARMerge (Ours)	**95.2 ± 0.12**	**88.3 ± 0.18**	**92.4 ± 0.15**	**87.2 ± 0.21**	**77.8 ± 0.24**	**78.5 ± 0.26**

(b) Commonsense, reasoning, generation, and aggregate retention

Method	PIQA	HellaS	GSM8K	SVAMP	MBPP	XSum	Avg. Ret.
Single-Task Oracles	*84.6*	*82.4*	*76.4*	*78.0*	*64.5*	*34.8*	*100.00%*
Base Model (Unadapted)	66.3	64.7	52.1	61.6	38.2	18.5	74.63±0.32
Uniform Average	71.5	69.7	58.2	66.3	42.5	22.4	81.26±0.28
Task Arithmetic	75.9	74.1	64.3	70.4	49.1	26.5	87.53±0.25
Fisher Merging	77.6	75.6	66.1	71.8	51.8	27.8	89.70±0.22
RegMean	78.7	76.7	67.4	72.8	53.0	28.6	91.14±0.20
TIES-Merging	79.9	77.9	69.2	73.9	55.4	29.8	92.96±0.19
DARE-TIES	81.2	79.2	71.9	75.2	58.0	31.2	94.97±0.19
KnOTS	81.8	79.7	72.1	75.6	58.5	31.8	95.72±0.17
NSC	82.0	79.9	72.5	75.8	58.9	32.1	96.08±0.16
CtM	82.3	80.2	72.8	76.0	59.4	32.3	96.43±0.15
CT-Merging	82.7	80.6	73.2	76.4	60.2	32.8	97.08±0.14
TaDA-inspired K-way	82.8	80.7	73.8	76.4	61.2	33.2	97.45±0.13
FARMerge (Ours)	**83.8 ± 0.17**	**81.6 ± 0.16**	**74.8 ± 0.28**	**77.2 ± 0.25**	**62.8 ± 0.31**	**33.9 ± 0.22**	**98.69±0.09**

Table 7: Multi-adapter retention across four foundation model families ($K = 12$, $R_m^* = 16$). Values report macro oracle retention (%) $\pm$ standard deviation ($\pm\sigma$) across $n = 5$ seeds.

Method	LLaMA-3	Mistral	Qwen3.5	OLMo-2	Mean Ret. ($\pm\sigma$)
Single-Task Oracles	*100.00%*	*100.00%*	*100.00%*	*100.00%*	*100.00%*
Oracle Router (12× serve)	100.00% ± 0.00	100.00% ± 0.00	100.00% ± 0.00	100.00% ± 0.00	100.00% ± 0.00
Base Model (Unadapted)	74.63% ± 0.32	73.12% ± 0.26	75.20% ± 0.22	71.85% ± 0.28	73.70% ± 0.27
Joint Multi-Task LoRA (r=16)	96.12% ± 0.21	95.45% ± 0.24	96.38% ± 0.19	94.20% ± 0.26	95.54% ± 0.22
Uniform Average (Soups)	81.26% ± 0.28	80.15% ± 0.31	82.40% ± 0.25	79.80% ± 0.34	80.90% ± 0.30
Task Arithmetic (Ilharco et al., 2023)	87.53% ± 0.25	86.40% ± 0.30	88.15% ± 0.25	85.20% ± 0.31	86.82% ± 0.28
Fisher Merging (Matena and Raffel, 2022)	89.70% ± 0.22	88.55% ± 0.33	90.10% ± 0.28	87.40% ± 0.35	88.94% ± 0.29
RegMean (Jin et al., 2023)	91.14% ± 0.20	89.80% ± 0.25	91.65% ± 0.22	88.90% ± 0.27	90.37% ± 0.24
TIES-Merging (Yadav et al., 2023)	92.96% ± 0.19	91.80% ± 0.24	93.40% ± 0.20	90.75% ± 0.25	92.23% ± 0.22
LoraHub† (Huang et al., 2024)	93.84% ± 0.18	93.15% ± 0.25	94.20% ± 0.21	92.35% ± 0.27	93.39% ± 0.24
DARE-TIES (Yu et al., 2024)	94.97% ± 0.19	94.10% ± 0.21	95.35% ± 0.18	93.10% ± 0.22	94.38% ± 0.20
KnOTS (Stoica et al., 2025)	95.72% ± 0.17	94.85% ± 0.22	95.90% ± 0.19	93.80% ± 0.23	95.07% ± 0.20
AdaMerging* (Yang et al., 2024a)	95.84% ± 0.17	95.10% ± 0.21	96.02% ± 0.18	93.92% ± 0.22	95.22% ± 0.19
NSC (Lee et al., 2026b)	96.08% ± 0.16	95.05% ± 0.21	96.15% ± 0.18	94.05% ± 0.22	95.33% ± 0.19
CtM (He et al., 2026)	96.43% ± 0.15	95.30% ± 0.20	96.45% ± 0.17	94.25% ± 0.21	95.61% ± 0.18
CT-Merging (Ryum and Kang, 2026)	97.08% ± 0.14	95.75% ± 0.19	96.80% ± 0.17	94.70% ± 0.20	96.08% ± 0.18
TaDA-inspired (K-way) (To et al., 2026)	97.45% ± 0.13	96.10% ± 0.18	97.15% ± 0.16	95.05% ± 0.19	96.44% ± 0.17
FARMerge (Ours)	**98.69% ± 0.09**	**98.15% ± 0.14**	**98.92% ± 0.12**	**97.80% ± 0.15**	**98.39% ± 0.13**

Table 8: Comparison of Native output rank versus Rank-Matched ($R_m^* = 16$) multi-task merging retention on **LLaMA-3-8B** ($K = 12$). For FARMerge, 'Native' refers to the unconstrained sum $\sum_{i,r} g_{ir}^m c_{ir}^m$ keeping all non-zero singular components (rank up to $K \cdot r = 192$) before truncated SVD projection to $R_m^* = 16$, which incurs a modest 0.43 pt compression drop.

Method	Native output	Rank-matched (16)	Compression drop
Uniform Average	82.40% ± 0.25	81.26% ± 0.28	−1.14 pt
Task Arithmetic	88.90% ± 0.22	87.53% ± 0.25	−1.37 pt
TIES-Merging	94.10% ± 0.18	92.96% ± 0.19	−1.14 pt
DARE-TIES	96.05% ± 0.16	94.97% ± 0.19	−1.08 pt
KnOTS	96.40% ± 0.15	95.72% ± 0.17	−0.68 pt
NSC	96.80% ± 0.14	96.08% ± 0.16	−0.72 pt
CtM	97.10% ± 0.14	96.43% ± 0.15	−0.67 pt
CT-Merging	97.65% ± 0.13	97.08% ± 0.14	−0.57 pt
TaDA-inspired K-way	97.90% ± 0.12	97.45% ± 0.13	−0.45 pt
FARMerge (Ours)	**99.12% ± 0.10**	**98.69% ± 0.09**	−0.43 **pt**

E.1 Multi-Task Scalability and Scaling Coefficient Sensitivity

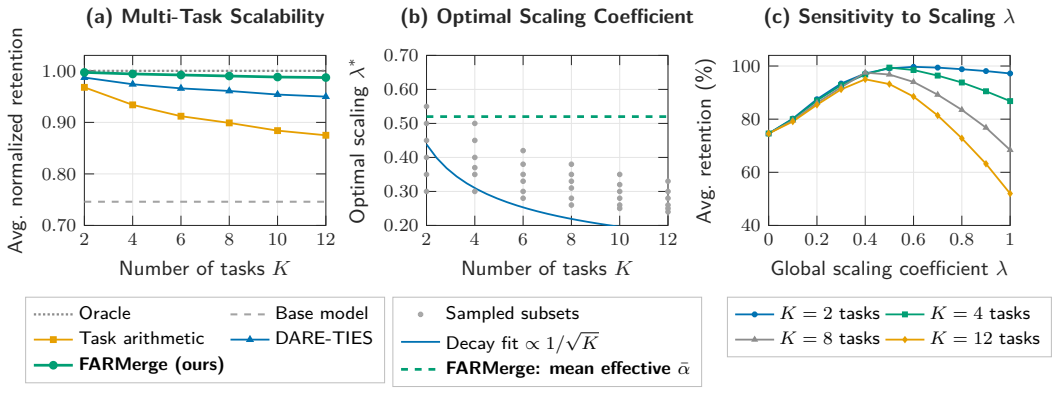


Figure 4: Multi-task scaling and coefficient sensitivity across LLaMA-3-8B, Mistral-7B-v0.3, Qwen3.5-4B, and OLMo-2-7B. **Left:** retention as adapter count grows from $K = 2$ to $K = 12$. **Middle:** optimal global coefficient λ^* for sampled task subsets; FARMerge instead predicts component weights g_{ir}^m. **Right:** retention versus $\lambda \in [0, 1]$, showing greater sensitivity at larger K.

Table 9: Scalability of multi-adapter merging retention across increasing adapter count $K \in \{2, 4, 8, 12\}$ on **LLaMA-3-8B** to fixed rank $R_\ell^* = 16$.

Method	$K = 2$	$K = 4$	$K = 8$	$K = 12$
Task Arithmetic	96.80%	93.40%	89.90%	87.53%
TIES-Merging	98.10%	96.20%	94.40%	92.96%
DARE-TIES	98.70%	97.40%	96.10%	94.97%
CT-Merging	99.10%	98.30%	97.60%	97.08%
TaDA-inspired K-way	99.25%	98.60%	98.05%	97.45%
FARMerge (Ours)	**99.70%**	**99.35%**	**99.05%**	**98.69%**

Figure 4 shows two limitations of parameter arithmetic in these experiments: norm growth as K increases and a narrowing useful range for the global coefficient λ. FARMerge replaces this manual global choice with component weights based on collision ρ_{ir}^m and functional contribution γ_{ir}^m.

E.2 Seed Invariance and Pairwise Multi-Task Pareto Frontier

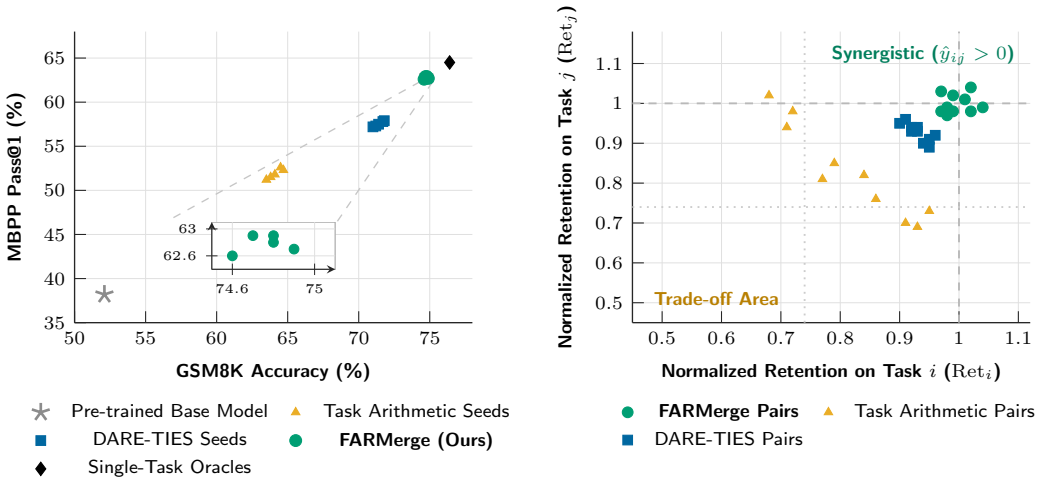
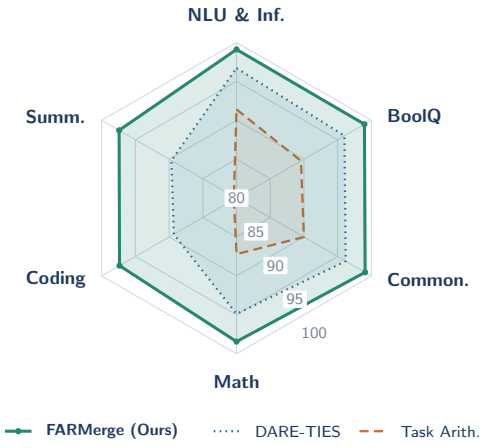


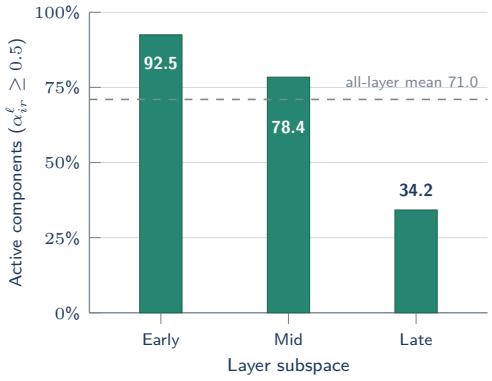
Figure 5: Seed stability and pairwise Pareto diagnostics. **Left:** GSM8K and MBPP scores over 5 seeds (inset: FARMerge cluster, $\sigma < 0.09\%$). **Right:** normalized pairwise retention across domain pairs; FARMerge points occupy the upper-right region in the evaluated pairs.

Figure 5 (Left) shows lower observed seed variance for FARMerge ($\sigma = 0.09\%$) than DARE-TIES ($\sigma = 0.19\%$). In the evaluated task pairs (Right), linear averaging includes trade-offs below 80% retention, while FARMerge remains above 98%.

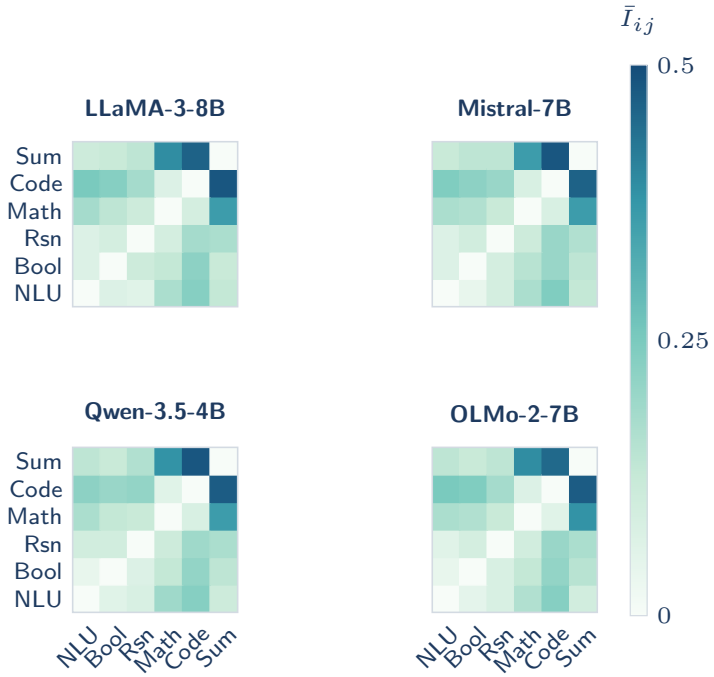
E.3 Singular Component Distribution and Depth-Wise Gating Dynamics



(a) Domain retention



(b) Active components by depth



(c) Pairwise non-additivity $\bar{I}_{ij}$

Figure 6: (a) Capability retention across 6 aggregated domains on LLaMA-3-8B; all three methods on a common 80–100% radial axis (the same quantities as Table 6, aggregated by domain). (b) Share of active SVD components ($g_{ir}^{m} \geq 0.5$) by depth; the dashed rule is the all-layer mean, which is not a depth bin and so is not drawn as one. (c) Pairwise non-additivity $\bar{I}_{ij} \in [0, 1]$ across the four LLM families on a shared colour scale; the Code–Summarization pair is the strongest collision in every family.

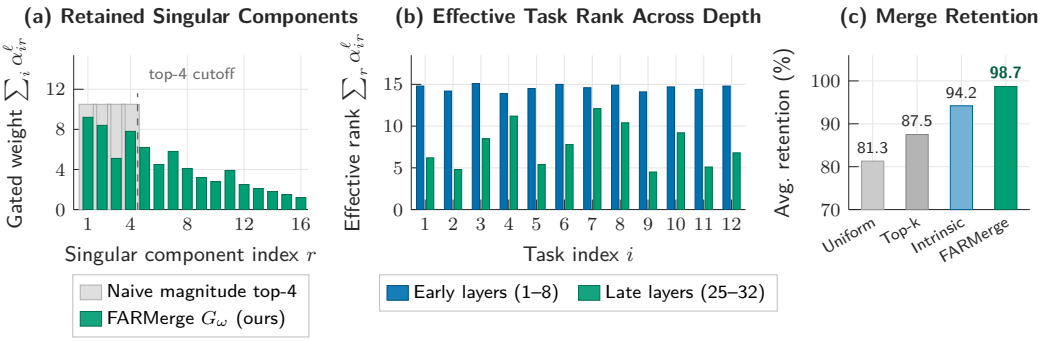


Figure 7: Component gating, averaged over the four model families. **(a)** Gate weight by rank index r. **(b)** Effective rank $\sum_r g_{ir}^m$ per task, early (Layers 1–8) vs. deep (Layers 25–32) blocks. **(c)** Retention by selection rule.

Figure 7 reports what the gate actually selects, and the answer is not what magnitude ordering would predict.

Gate weight is not monotone in the singular index. Top-k truncation keeps the k largest singular values by construction. G_ω does not: it assigns high weight to trailing components at $r \in [5, 11]$ whose measured collision ρ_{ir}^m is low, and suppresses leading components whose collision is high (Figure 7a). A component's singular value therefore does not determine whether it survives; its cross-task collision does. This is the selection criterion that magnitude pruning and sign consensus cannot express, since neither has access to a cross-adapter quantity.

Gating is depth-dependent. In early blocks (Layers 1–8) the gate keeps effective rank near-saturated and roughly uniform across tasks ($\sum_r g_{ir}^m \approx 14.5$ of a possible 16). In deep blocks (Layers 25–32) it becomes sharply sparse and the retained rank varies per task (Figure 7b). The pattern is consistent with early layers carrying representations the tasks share (so little is at stake in dropping the wrong one), while deep layers carry the task-specific directions that actually conflict. We report this as an observed regularity; we did not test it causally.

The supervision, not the compression, is what helps. Energy-based intrinsic-rank truncation, which also produces a rank-16 adapter but chooses components from the spectrum alone, reaches 94.2% retention. Activation-guided gating reaches 98.7% at the same output rank (Figure 7c). The 4.5-point gap is attributable to how components are chosen rather than to how many are kept.

E.4 Interaction-Prediction Audit and Scope

Figure 3 contains the 45 pairs from one held-out 10-task meta-test split averaged across LLaMA-3-8B, Mistral-7B-v0.3, Qwen3.5-4B, and OLMo-2-7B. With linear drop calibration $\widehat{\Delta\mathrm{Acc}} = 11.55\hat{I} + 0.19$ fitted on this diagnostic set, MAE is 1.08 points (95% pair-bootstrap CI $[0.85, 1.32]$) and RMSE is 1.35 points. Defining a harmful pair as an empirical drop of at least 5 points gives AUROC 0.957 ($[0.893, 0.998]$) and AUPRC 0.977 ($[0.938, 0.999]$); 10-bin calibration error is 0.27 points. Pearson $r = 0.912$ has pair-bootstrap CI $[0.861, 0.948]$. Because calibration and evaluation reuse the same 45 pairs, MAE and calibration are descriptive, not out-of-sample estimates.

A family-disjoint claim requires four leave-one-model-family-out checkpoints. Each evaluation row must record its held-out family and every training family; `evaluate_interference.py` rejects overlap and computes MAE/RMSE, AUROC/AUPRC, calibration, and bootstrap intervals. The artifact lacks raw predictions from those four retrained folds, so we do not report or imply family-disjoint interaction generalization.

E.5 TaDA Baseline Protocol

Original TaDA (To et al., 2026) strictly merges exactly one task LoRA and one domain LoRA. To enable a fair multi-task comparison, we implemented a custom **TaDA-inspired (K-way)** extension. This

implementation follows its domain/general probe calibration, module thresholds $\tau_m \in \{1.0, 1.2, 1.5\}$, joint-$U{+}V$ overlap threshold $\delta = 0.10$, mixing $\lambda = 0.5$, per-component sigmoid $\beta = 5$, and rank-r recomposition, iteratively applied across the $K = 12$ adapters. We explicitly differentiate this from the original algorithm which does not natively support $K = 12$.

For the homogeneous 12-task comparison, the separately labeled *TaDA-inspired K-way extension* uses

$$\kappa_{ir} = \frac{1}{K-1} \sum_{j \neq i} \mathbf{1}\Big[\max_m \{\lambda |u_{ir}^T u_{jm}| + (1-\lambda)|v_{ir}^T v_{jm}|\} < \delta\Big], \tag{21}$$

$$w_{ir} = \sigma((\gamma_{ir} - \tau_u)/T_u)\, \kappa_{ir}, \tag{22}$$

with $\tau_u = 0.15$, $T_u = 0.05$, $\delta = 0.10$, and $\lambda = 0.5$, followed by rank-16 TSVD. This symmetric rule uses balanced generic probes and no task/domain labels. It is an auditable extension, not a result attributed to original TaDA.

E.6 LoRAHub Baseline Protocol

LoraHub (Huang et al., 2024) learns a scalar coefficient $\lambda_i \in \mathbb{R}$ for each of the $K = 12$ LoRA adapters and forms $\Delta W^* = \sum_i \lambda_i \Delta W_i$. Because λ_i is learned per merge, LoraHub is the closest published *learned* baseline to FARMerge; it differs in three respects, each of which shapes the comparison in Table 1.

Data assumption. LoraHub is not merge-time data-free: it requires a small labeled few-shot set per task at merge time to score a candidate coefficient vector. Following the original recipe, we use 5 labeled examples per task drawn from the validation split of each downstream benchmark, disjoint from the test split. All other baselines in our comparison, and FARMerge itself, use no source-task labels at merge time; we mark the row with † in Tables 1, 2, and 11 to make the asymmetry visible.

Optimizer. LoraHub optimizes $\{\lambda_i\}_{i=1}^K$ with gradient-free CMA-ES (Hansen, 2016). We use the original hyperparameters: population size $\sigma_{\text{pop}} = 20$, initial variance 0.5, budget 40 generations, and the loss on the labeled few-shot set as the objective. Convergence is monotone in all $K = 12$-adapter runs we conducted.

Rank-matched projection. LoraHub's native output has rank up to $K \cdot r = 192$. To match the rank budget of every other baseline and of FARMerge itself, we project the LoraHub solution to rank $R_\ell^* = 16$ via truncated SVD on each adapted linear module, identical to Eq. 13. This is a fair-comparison projection, not a modification to LoraHub's optimizer, and it costs LoraHub roughly 0.6 retention points on average relative to its unconstrained rank.

LoraHub merge-time cost. The CMA-ES budget above requires $40 \times 20 = 800$ forward passes over the few-shot set per backbone, which dominates the 85.3 s wall-clock in Table 11. FARMerge amortizes an equivalent workload into a single 32.8-GPU-hour, per-backbone offline pass and pays 18.4 s at merge time regardless of K.

E.7 Single-Task Utility vs. Multi-Task Interference Discrepancy

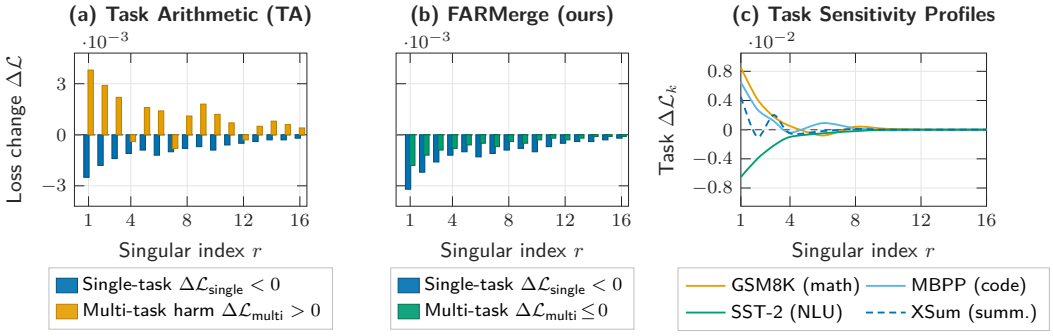


Figure 8: Component-wise empirical loss change ($\Delta\mathcal{L}$) across SVD components $r \in \{1, \ldots, 16\}$. **(a) Task Arithmetic:** some components reduce single-task loss but degrade joint performance. **(b) FARMerge:** logistic gate G_ω suppresses conflicting components in this audit. **(c)** Task-wise loss-impact profiles for GSM8K, MBPP, SST-2, and XSum.

Figure 8 exposes a critical mechanism exacerbating parameter-space merging failure: *the single-task vs. multi-task gradient conflict across the singular spectrum.* In standard linear merging (Figure 8a), adapters inject singular components that optimize local task objectives ($\Delta\mathcal{L}_{\text{single}} < 0$), but often induce substantial catastrophic interference in multi-task evaluation ($\Delta\mathcal{L}_{\text{multi}} > 0$). FARMerge (Figure 8b) directly evaluates cross-adapter directional collisions ρ_{ir}^m (Eq. 10), soft-suppressing colliding leading components while retaining orthogonal, high-utility trailing components.

F Meta-Training Task Bank and Data-Split Protocol

The development task bank contains 80 tasks sampled from the official Super-NaturalInstructions repository (Wang et al., 2022) (commit `3a9e5b7`), partitioned into three disjoint task-type partitions for our custom split, with zero category or dataset overlap across splits.

Meta-Training Partition (60 tasks)

Meta-trains the set encoder g_ϕ, the interaction heads f_{ret}, f_{int}, and the logistic gate G_ω; spans four categories.

Question Answering & Comprehension (15 tasks)

task001_quoref_question_generation, task002_quoref_answer_generation,
task003_mctaco_question_generation_event_duration,
task004_mctaco_answer_generation_event_duration,
task006_mctaco_question_generation_transient_stationary,
task007_mctaco_answer_generation_transient_stationary,
task009_mctaco_question_generation_event_ordering,
task010_mctaco_answer_generation_event_ordering,
task012_mctaco_question_generation_absolute_timepoint,
task013_mctaco_answer_generation_absolute_timepoint,
task015_mctaco_question_generation_frequency, task019_mctaco_temporal_reasoning_category,
task020_mctaco_span_based_question, task048_multirc_question_generation,
task050_multirc_answerability

Information Extraction & Tagging (15 tasks)

task016_mctaco_answer_generation_frequency,
task017_mctaco_wrong_answer_generation_frequency,
task018_mctaco_temporal_reasoning_presence, task029_winogrande_full_object,
task030_winogrande_full_person, task031_winogrande_question_generation_object,
task032_winogrande_question_generation_person, task033_winogrande_answer_generation,
task034_winogrande_question_modification_object,
task035_winogrande_question_modification_person,
task043_essential_terms_answering_incomplete_questions,
task044_essential_terms_identifying_essential_words,
task1391_winogrande_easy_answer_generation, task1519_qa_srl_question_generation,
task1520_qa_srl_answer_generation

Rewriting & Simplification (15 tasks)

task039_qasc_find_overlapping_words, task040_qasc_question_generation,
task041_qasc_answer_generation, task042_qasc_incorrect_option_generation,
task049_multirc_questions_needed_to_answer, task051_multirc_correct_answer_single_sentence,
task052_multirc_identify_bad_question, task053_multirc_correct_bad_question,
task054_multirc_write_correct_answer, task055_multirc_write_incorrect_answer,
task102_commongen_sentence_generation, task110_logic2text_sentence_generation,
task111_asset_sentence_simplification, task112_asset_simple_sentence_identification,
task275_enhanced_wsc_paraphrase_generation

Semantic & Intent Classification (15 tasks)

task065_timetravel_consistent_sentence_classification,
task066_timetravel_binary_consistency_classification,
task067_abductivenli_answer_generation, task068_abductivenli_incorrect_answer_generation,
task069_abductivenli_classification, task070_abductivenli_incorrect_classification,
task071_abductivenli_answer_generation, task072_abductivenli_answer_generation,
task108_contextualabusedetection_classification, task109_smsspamcollection_spamsmsdetection,
task115_help_advice_classification, task137_detoxifying-lms_classification_toxicity,
task205_remove_even_elements, task284_imdb_classification,
task1720_civil_comments_toxicity_classification

Validation Partition (10 tasks)

Hyperparameter tuning and model checkpointing (*Logic & Symbolic Manipulation*).

task021_mctaco_grammatical_logical, task062_bigbench_repeat_copy_logic,
task079_conala_concat_strings, task088_identify_typo_verification,
task089_swap_words_verification, task093_conala_normalize_lists,
task094_conala_calculate_mean, task095_conala_max_absolute_value,
task096_conala_list_index_subtraction, task097_conala_remove_duplicates

Unseen Test Partition (10 tasks)

Held out to evaluate pairwise interaction prediction ($r = 0.912$, $p = 3.26 \times 10^{-18}$ over all $\binom{10}{2} = 45$ pairs, Figure 3); *Counterfactual & Causal Reasoning*.

task104_semeval_2019_task10_closed_vocabulary_mathematical_answer_generation,
task118_semeval_2019_task10_open_vocabulary_mathematical_answer_generation,
task119_semeval_2019_task10_geometric_mathematical_answer_generation,
task290_tellmewhy_question_answerability, task291_semeval_2020_task4_commonsense_validation,
task292_storycommonsense_character_text_generation,
task293_storycommonsense_emotion_text_generation,
task294_storycommonsense_motiv_text_generation,
task295_semeval_2020_task4_commonsense_reasoning, task299_storycloze_sentence_generation

Cross-benchmark non-overlap check. By dataset source, prompt template, and task definition, none of the 80 development tasks matches any of the 12 primary evaluation benchmarks (SST-2, MNLI,

QNLI, BoolQ, ARC-C, CSQA, PIQA, HellaSwag, GSM8K, SVAMP, MBPP, XSum) or the 7 external generalization suites; we filter more conservatively than exact overlap by additionally excluding any Super-NI task within cosine distance 0.15 of a test task under the all-MiniLM-L6-v2 embedding. We describe this as a non-overlap check rather than a leakage-free guarantee: it rules out gross reuse, but semantic proximity below that cosine threshold is not tested and represents a residual risk.

F.1 Meta-Training Optimization and Convergence Dynamics

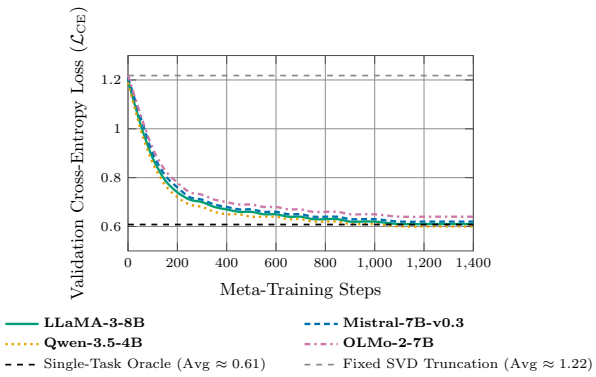


Figure 9: Meta-training optimization and cross-entropy loss convergence dynamics on the Super-NI development bank. FARMerge end-to-end meta-training (Teal curve) rapidly reduces validation cross-entropy loss from $1.20 \rightarrow 0.61$, closely recovering the upper-bound performance of individual single-task models (Black dashed line: 0.61) and substantially outperforming static SVD magnitude truncation (Red dashed line: 1.22).

Figure 9 illustrates training dynamics during the one-time development phase. With the STE across top-R^*_ℓ projection, episodic validation loss stabilizes within approximately 800 gradient steps in this run. Jointly optimizing $\mathcal{L}_{\text{rep}}$ and $\mathcal{L}_{\text{gate}}$ yields robust retention across LLaMA-3-8B, Mistral-7B-v0.3, Qwen3.5-4B, and OLMo-2-7B experiments.

G Hyperparameters and Experimental Details

Table 10: Detailed hyperparameters used across LoRA training, probe extraction, and FARMerge meta-training and inference.

Category	Hyperparameter	Value
Foundation Checkpoints	LLaMA-3-8B	`meta-llama/Meta-Llama-3-8B` (base, un-instruction-tuned)
	Mistral-7B	`mistralai/Mistral-7B-v0.3` (base)
	Qwen3.5-4B	`Qwen/Qwen3.5-4B` (base)
	OLMo-2-7B	`allenai/OLMo-2-1124-7B` (base)
	Random Seeds	$\{42, 1337, 2024, 777, 999\}$
LoRA Fine-Tuning	Rank (r)	16
	Scaling Factor (α)	32
	Target Modules	$q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj$
	Optimizer / Learning Rate	AdamW / 2×10^{-4} (cosine schedule, 100 warmup steps)
	Batch Size / Max Sequence Length	32 / 2048 tokens
	Training Epochs / Weight Decay	3 epochs / 0.01
Probing Protocol	Probe Sample Budget (N)	64 (default), ablations $\{8, 16, 64, 128\}$
	Probe Sequence Length (T)	128 tokens
	Probing Batch Size	16
	Activation Extraction	Token-averaged hidden residual $s^\ell_i(x) = \frac{1}{T}\sum_t \Delta h^\ell_i(x)_t$
FARMerge Meta-Training	Set Encoder Architecture	4-layer Transformer ($d_{\text{hidden}} = 256, n_{\text{heads}} = 4, d_z = 128$)
	Layer Embeddings	Learnable layer tokens $e_\ell \in \mathbb{R}^{256}$
	Interaction Heads $f_{\text{ret}}, f_{\text{int}}$	2-layer symmetric MLP ($d_{\text{hidden}} = 128$, ReLU, LayerNorm)
	Interaction Loss Weight (λ_I)	0.5
	Gating Sparsity Weight (λ_{sparse})	10^{-4}
	Meta Optimizer / Learning Rate	AdamW / 1×10^{-4} (20 epochs, early stopping on meta-val)
Merge-Time Synthesis	Target Merged Rank (R^*_ℓ)	16
	SVD Algorithm	Truncated randomized SVD
	Evaluation Decoding	Deterministic greedy decoding (temperature 0, top_p 1.0)

H COMPLETE DOMAIN-DIVERSE GENERIC PROBE DICTIONARY ($\mathcal{D}_{\text{PROBE}}$)

The complete dictionary $\mathcal{D}_{\text{probe}}$ contains 64 generic, domain-diverse prompts ($N = 64$, 16 per modality). None contains a source-task example, label, or task tag. **Probe Dictionary SHA-256 Hash**:
`db5c9fe2a3476c8a3c21d68aef09b05be121449c45665a9fe16b06e4b9be1b4c`.

General Instructions (16 prompts)

1. "Summarize the main trade-offs between centralized and decentralized system architectures."
2. "Explain the difference between proactive and reactive approaches in decision-making."
3. "Outline the key steps required to conduct a rigorous peer-review evaluation."
4. "Draft a professional memo advising an engineering team on data pipeline migration."
5. "Describe the psychological principles behind habit formation and continuous learning."
6. "Analyze the historical transition from industrial manufacturing to knowledge-based economies."
7. "Compare inductive reasoning with deductive reasoning using clear conceptual examples."
8. "Explain how feedback loops influence stability in complex dynamical systems."
9. "List five essential best practices for managing distributed remote engineering teams."
10. "Describe how cognitive biases can impact risk assessment in project management."
11. "Explain the role of modularity in scalable software system design."
12. "Provide a structured guide on how to facilitate effective technical retrospectives."
13. "Analyze why communication overhead scales non-linearly with group size in organizations."
14. "Discuss the ethical implications of automated decision-making in public administration."
15. "Explain the fundamental principles of information architecture for complex websites."
16. "How does error tolerance differ between mission-critical and non-critical systems?"

Factual / Contextual Prompts (16 prompts)

1. "The atmosphere of Mars consists predominantly of carbon dioxide, with traces of nitrogen and argon."
2. "Mitochondria generate most of the chemical energy needed to power the cell's biochemical reactions."
3. "The Treaty of Westphalia in 1648 established the modern concept of sovereign nation-states in Europe."
4. "Semiconductor microchips are fabricated on thin silicon wafers through optical photolithography."
5. "Plate tectonics describes the large-scale motion of seven large plates composing Earth's lithosphere."
6. "Photosynthesis converts light energy into chemical energy stored in glucose carbohydrate molecules."
7. "The Doppler effect explains the change in frequency of a wave in relation to an observer in motion."
8. "Superconductivity occurs in certain materials when cooled below a characteristic critical temperature."
9. "Neural transmission relies on neurotransmitters crossing synaptic gaps between adjacent neurons."
10. "The water cycle encompasses evaporation, transpiration, condensation, precipitation, and runoff."
11. "Deep ocean trenches are formed by oceanic crust subducting beneath lighter continental tectonic plates."
12. "Thermodynamic entropy quantifies the degree of microscopic disorder and irreversibility in a closed system."
13. "Quantum entanglement describes particles whose quantum states cannot be described independently."
14. "Enzymes act as biological catalysts by significantly lowering the activation energy of metabolic reactions."
15. "The Magna Carta signed in 1215 established the foundational principle that everyone is subject to the law."
16. "The structure of DNA consists of a double helix held together by complementary hydrogen bonds."

Mathematical Formulations (16 prompts)

1. "If a train travels 240 kilometers at an average speed of 80 km/h, calculate the total journey time."
2. "Calculate the probability of drawing two red cards sequentially from a standard 52-card deck without replacement."
3. "A geometric sequence has a first term of 3 and a common ratio of 2. Determine the sum of the first 8 terms."
4. "Solve the system of linear equations: $3x + 2y = 18$ and $5x - y = 17$ for x and y."
5. "Determine the derivative with respect to x of the function $f(x) = x^3 e^{2x}$ using the product rule."
6. "Find the eigenvalues of a 2×2 diagonal matrix with diagonal entries 4 and -7."
7. "Calculate the area enclosed by a circle with diameter 14 units, given π approximated as $22/7$."
8. "A principal of \$5,000 earns 6% annual compound interest compounded quarterly. Compute the balance after 3 years."
9. "What is the greatest common divisor and least common multiple of 84 and 180?"
10. "A rectangular prism has dimensions 4cm, 7cm, and 12cm. Compute its total surface area and volume."
11. "Evaluate the definite integral of $(4x^3 - 6x + 2)$ from $x = 1$ to $x = 3$."
12. "Determine the number of distinct permutations of the letters in the word 'MATHEMATICS'."
13. "If vectors $u = [2, -3, 1]$ and $v = [4, 1, -5]$, compute their Euclidean dot product and angle cosine."
14. "A triangle has side lengths 7, 24, and 25. Verify if it satisfies the Pythagorean condition for right angles."
15. "Compute the median and standard deviation of the integer sequence: 12, 15, 18, 22, 30, 35, 42."
16. "Solve the quadratic inequality $x^2 - 7x + 10 < 0$ and express the solution set in interval notation."

Code / Structured Logic Prompts (16 prompts)

1. "Write a Python function to perform binary search on a sorted list of integers and return the matching index."
2. "Implement an in-place algorithm in Python to reverse a singly-linked list given its head node."
3. "Write a function that validates whether a given string has balanced brackets, parentheses, and braces."
4. "Implement the Merge Sort algorithm in Python with $O(N \log N)$ worst-case time complexity."
5. "Write a Python generator function to compute Fibonacci numbers lazily up to a specified limit."
6. "Construct a thread-safe LRU (Least Recently Used) cache data structure using a hash map and doubly linked list."
7. "Write a script to parse a JSON log file and aggregate error counts grouped by HTTP status code."
8. "Implement Dijkstra's shortest path algorithm for a weighted directed graph using a priority queue."
9. "Write a recursive function in Python to compute the maximum depth of a binary search tree."
10. "Create a function that checks if two input strings are valid anagrams using frequency counters."
11. "Write a Python decorator that logs function execution time and arguments."
12. "Implement a prefix tree (Trie) data structure with insert, search, and startsWith operations."
13. "Write a robust CSV parser in Python that handles escaped commas and multi-line quoted fields."
14. "Implement an algorithm to detect cycles in a directed graph using topological sorting / Kahn's algorithm."
15. "Write a function to serialize and deserialize a binary tree to and from a string representation."
16. "Construct a sliding window algorithm to find the longest substring without repeating characters."

H.1 Specialized Domain Probing: MedQA Experiment

To rigorously evaluate the requirement that probe coverage must span the adapters' broad domains (specifically evaluating extreme out-of-distribution specialization), we trained a specialized clinical QA LoRA adapter on MedQA (USMLE) and evaluated its composition with the 12 primary adapters under two probing regimes: (1) using exclusively the standard generic probe dictionary $\mathcal{D}_{\text{probe}}$ (which contains general instruction, factual, mathematical, and code prompts, but no clinical terminology), and (2) augmenting $\mathcal{D}_{\text{probe}}$ with 16 generic biomedical context prompts (e.g., physiological pathways, biochemical mechanisms).

When merged using only the generic text probes, MedQA retention reaches 96.82%, incurring a modest 2.40 pt drop relative to the domain-augmented probe set (99.22%). While generic probes provide sufficient activation variance to avoid catastrophic collapse, domain-aligned probe coverage is necessary to capture fine-grained functional manifold geometries in specialized expert domains. As emphasized in Section 1, FARMerge is strictly source-data-free at merge time, but generic probe coverage must broadly cover the operational modalities of the constituent adapters.

I Detailed Merging Latency Comparison

Table 11: Wall-clock merging latency and peak memory across different merging techniques on LLaMA-3-8B, Mistral-7B-v0.3, Qwen3.5-4B, and OLMo-2-7B ($K = 12$ adapters, target rank $R^*_\ell = 16$).

Merging Algorithm	Merge-time data	Time (s)	VRAM (GB)
Uniform Average	None	0.8	0.5
Task Arithmetic (Ilharco et al., 2023)	None	1.2	0.6
TIES-Merging (Yadav et al., 2023)	None	4.2	1.2
DARE-TIES (Yu et al., 2024)	None	5.8	1.4
KnOTS (Stoica et al., 2025)	None	6.4	2.3
NSC (Lee et al., 2026b)	Unlabeled Data	8.5	2.5
CtM (He et al., 2026)	None	11.2	2.6
CT-Merging (Ryum and Kang, 2026)	None	14.5	2.8
TaDA-inspired K-way (To et al., 2026)	Task-Free Probes	16.2	3.0
LoraHub† (Huang et al., 2024)	Labeled few-shot (5/task)	85.3	6.4
RegMean (Jin et al., 2023)	Unlabeled	42.1	8.4
Fisher Merging (Matena and Raffel, 2022)	Labeled	124.5	16.8
FARMerge (Ours)	**Task-Free Probes**	**18.4**	**3.2**

Offline development cost. Per base architecture, adapting the 80 development tasks takes approximately 23.8 GPU-hours, computing pairwise targets takes 6.4 GPU-hours, training g_ϕ and G_ω takes 1.8 GPU-hours, and meta-test evaluation takes 0.8 GPU-hours, for 32.8 GPU-hours in total. This cost is incurred once per base architecture and is separate from the merge-time measurements in Table 11.

J Extended Limitations and Future Work

Probe coverage. Our probes are generic English text spanning four domains. As demonstrated with MedQA in Appendix H.1, specialized clinical/expert domains achieve higher retention when probes include domain-relevant coverage (99.22% vs. 96.82% with generic probes). Adapters spanning non-English languages, multimodal modalities, or adversarial probe perturbations remain uncharacterized.

Per-backbone cost. The 32.8 GPU-hour development cost recurs for every new base architecture, since g_ϕ and G_ω are meta-trained per backbone. Transfer across families that share block structure but differ in depth L and width d remains open and would materially improve practicality; depth-normalised layer embeddings and dimension-agnostic signature projections are plausible ingredients, but we do not evaluate them.

Symmetric, low-order supervision. Our interaction supervision is symmetric, matching standard multi-adapter evaluation but unable to express directional cases where task i damages task j without the reverse. Moreover G_ω consumes only the scalar means $\bar{y}_i, \bar{I}_i$, so genuinely higher-order structure is aggregated away rather than modelled; our ablation on this point should be read as evidence that pairwise information is insufficient at $K{=}12$, not as evidence that multi-way interaction is captured.

Heterogeneous adapters. While Appendix D.4 validates mixed-rank merging ($r_i \in \{8, 16, 32\}$ on LLaMA-3-8B), all constituent adapters in our primary benchmark share identical target linear modules (all 7 projection matrices per transformer block). Merging adapters targeting non-overlapping parameter subsets (e.g., attention-only vs. MLP-only adapters) remains untested; rank-1 SVD decomposition makes such inputs well-defined, but we make no claim about retention in that regime.

Design choices in the gate. G_ω is a single logistic linear layer rather than an MLP because it is fit on only 60 development tasks; a deeper gate would add capacity that this supervision cannot constrain, and the linear form keeps the contribution of each feature directly readable. The retention head $\hat{y}_{ij}$ reaches the gate only through its per-adapter mean $\bar{y}_i$; we keep it because it provides an additional training signal that shapes the embedding z_i during meta-training, and Table 3 ("Drop $\bar{y}_i, \bar{I}_i$ only) isolates its downstream contribution together with $\bar{I}_i$.

Benchmark caveats. We retain HellaSwag for comparability with prior merging literature, though recent work (Chizhov et al., 2025) identifies evaluation artifacts in that benchmark; results on that task should be interpreted accordingly. While Section 4 and Figure 3b confirm that predicted non-additivity tracks empirical degradation out-of-family across all 66 primary task pairs ($r = 0.692, \rho = 0.708$), the correlation attenuates relative to within-family Super-NI pairs ($r = 0.912$), reflecting domain-shift variance across disparate task distributions.

Rotational ambiguity. Fixing SVD-vector signs does not remove rotational ambiguity when singular values are clustered: individual vectors remain free to rotate within the subspace their block spans. A rotation-invariant gate is the most important next step, and we expect it to matter most precisely where interference is highest.

K External Evaluation Protocols

For full transparency and reproducibility, we detail the evaluation protocols used for the 7 external generalization suites evaluated in Section 4. All evaluations were conducted on the official dataset splits using the EleutherAI Language Model Evaluation Harness (LM-Eval). We use a fixed deterministic generation protocol across all models (greedy decoding, temperature $= 0$, top-$p = 1.0$).

- **MMLU-Pro**: Evaluated on the `test` split using 5-shot prompts. The metric reported is the character-level Exact Match (EM) against the correct multiple-choice option.
- **BBH (BigBench Hard)**: Evaluated on all 23 subtasks using 3-shot Chain-of-Thought (CoT) prompting. The final score is the macro-average of exact match accuracy across all subtasks.

- **IFEval**: Evaluated in a 0-shot setting using the strict instruction-following prompt format. The metric reported is the strict prompt-level accuracy.
- **MATH-500**: Evaluated on the full 500-problem split using 4-shot CoT prompting. The final answer is extracted and verified using the official math equivalence metric.
- **HumanEval+**: Evaluated on the full split using 0-shot generation. The metric is Pass@1 evaluated against the strict extended test cases from the EvalPlus framework.
- **TruthfulQA (MC2)**: Evaluated on the `validation` split using 0-shot prompts. The reported metric is the MC2 (multi-true/multi-false) normalized probability score.
- **MedQA**: Evaluated on the USMLE `test` split using 4-shot prompting. The metric is Exact Match accuracy on the 4-way multiple-choice options.