GATED TASK ARITHMETIC: CONTROLLING TASK-SPECIFIC INPUT INFORMATION FLOW IN MODEL MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model merging has emerged as an effective approach for consolidating multiple task-specific fine-tuned models into a single unified model. Among existing approaches, data-free merging is particularly attractive as it does not require access to the original training data. A central challenge in model merging is to preserve task-specific knowledge while mitigating interference among task vectors. However, existing merging methods mainly manipulate task vectors themselves, without explicitly controlling the input representations on which they operate. After merging, each task vector may therefore operate on input representations associated with other tasks, potentially introducing cross-task interference. In this work, we propose **Gated Task Arithmetic (GTA)**, a data-free model merging method that introduces a task-specific input gate for each task vector. GTA constructs a gated task vector as $\widetilde{\tau}_t = \tau_t G_t$ and performs task arithmetic over the resulting gated task vectors. We design G_t as a **task-preserving and interference-suppressing gate (PSI-Gate)**, which preserves input representations associated with its corresponding task while suppressing those associated with other tasks. To learn the PSI-Gates without the original training data, we introduce a **task-preserving and interference-suppressing surrogate loss (PSI surrogate loss)** using task-vector-derived surrogate input representations. Under task-vector Gram covariance estimates and the stated convergence conditions, finite-step gate optimization from identity induces a preservation-oriented spectral bias, while its limiting solution corresponds to closed-form activation-aware merging. We extensively evaluate GTA across ViT models with varying numbers of vision tasks, as well as LLaMA and Qwen2 models fine-tuned on diverse natural language tasks. GTA consistently achieves strong merging performance across different model architectures, scales, and modalities, improving the recovery rate of task-vector performance by more than 16 percentage points over existing merging methods in the best case.

1 INTRODUCTION

With the emergence of foundation models, adapting large-scale pretrained models to diverse downstream tasks has produced numerous task-specific models, making their separate storage, deployment, and management increasingly inefficient (Bommasani et al., 2022). Model merging offers an effective alternative by constructing a single model without additional data-intensive training (Matena & Raffel, 2022; Wortsman et al., 2022), typically by aggregating "task vectors," defined as the parameter differences between a shared base model and its fine-tuned variants (Ilharco et al., 2023). However, task vectors learned from different tasks may conflict with one another, as their parameter updates can differ in both direction and magnitude. Such conflicts give rise to *task interference*, a key challenge in model merging that can substantially degrade the performance of the merged model.

To address task interference, extensive work has been conducted on model merging. Pruning-based methods remove or rescale interfering components of task vectors (Yu et al., 2024; Yadav et al., 2023; Deep et al., 2024). Singular value decomposition (SVD)-based methods mitigate interference by orthogonalizing task-vector subspaces or adjusting their singular values (Gargiulo et al., 2025;

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.

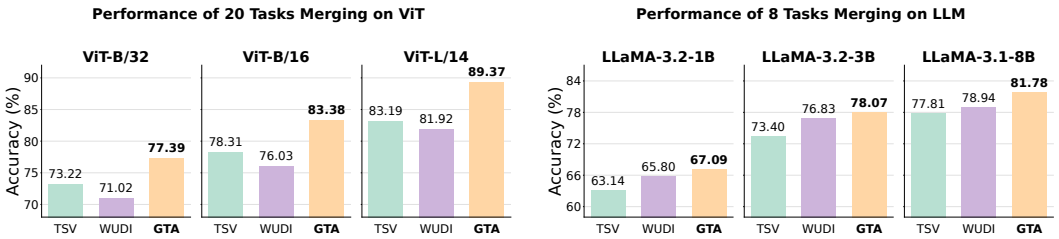


Figure 1: Merging performance across vision and language models using three ViT model sizes on 20 vision tasks (left) and three LLaMA model sizes on 8 natural language classification tasks (right). GTA consistently outperforms TSV-Merging and WUDI-Merging across all six model configurations.

Marczak et al., 2025; Skorobogat et al., 2026). Optimization-based methods formulate explicit merging objectives and optimize merging coefficients, task vectors, or correction directions (Yang et al., 2024b; Xiong et al., 2025; Wei et al., 2025; Cheng et al., 2025).

However, existing merging methods mainly manipulate task vectors themselves, without explicitly considering the input representations on which they operate. A task vector $\boldsymbol{\tau}_t$ is originally optimized for its corresponding task and therefore specializes in the input representations of that task. After merging, however, each task vector may also operate on input representations associated with other tasks, potentially introducing cross-task interference.

To directly control the input information supplied to each task vector, we propose **Gated Task Arithmetic (GTA)**, which introduces a task-specific input gate $\boldsymbol{G}_t$ for each task vector $\boldsymbol{\tau}_t$. Specifically, GTA constructs a gated task vector as $\widetilde{\boldsymbol{\tau}}_t = \boldsymbol{\tau}_t \boldsymbol{G}_t$, and merges the resulting gated task vectors as $\boldsymbol{\tau}_{\mathrm{GTA}} = \sum_{t=1}^{T} \widetilde{\boldsymbol{\tau}}_t = \sum_{t=1}^{T} \boldsymbol{\tau}_t \boldsymbol{G}_t$. We design $\boldsymbol{G}_t$ as a **task-preserving and interference-suppressing gate (PSI-Gate)**, which preserves task-relevant input information while suppressing the influence of input representations associated with other tasks. Consequently, each task vector selectively operates on its task-relevant input information, reducing cross-task interference during merging.

We further introduce a **task-preserving and interference-suppressing surrogate loss (PSI surrogate loss)** to learn each PSI-Gate $\boldsymbol{G}_t$ without access to the original training data. The PSI surrogate loss encourages two complementary properties: task preservation, $\boldsymbol{G}_t \boldsymbol{H}_t \approx \boldsymbol{H}_t$, and interference suppression, $\boldsymbol{G}_t \boldsymbol{H}_j \approx \boldsymbol{0}$ for $j \neq t$, where $\boldsymbol{H}_i$ denotes a batch of input representations from task i.

Our contributions are summarized as follows:

1. **Gated Task Arithmetic (GTA)** - We formulate model merging through task-vector-specific input information flow by equipping each task vector with an independent PSI-Gate and merging the gated vectors as $\boldsymbol{\tau}_{\mathrm{GTA}} = \sum_t \boldsymbol{\tau}_t \boldsymbol{G}_t$.
2. **Data-free PSI-Gate learning with finite-step bias** - We derive surrogate input representations directly from task vectors and learn PSI-Gates with the PSI surrogate loss, which promotes task preservation and interference suppression. Under task-vector Gram covariance estimates and the stated convergence conditions, the closed-form ACTMat solution (Hameed et al., 2026) corresponds to GTA's $K \to \infty$ limit, whereas finite-step optimization from identity retains a preservation-oriented spectral bias.
3. **Strong performance across vision and language models** - We extensively evaluate GTA across ViT, LLaMA, and Qwen2 models with varying model sizes and numbers of merged tasks. GTA achieves strong performance across vision and language settings, improving the recovery rate of task-vector performance by more than 16 percentage points in the best case.

2 Related Work

Averaging and weighted-sum-based merging. Early methods combine fine-tuned models through parameter averaging or weighted summation (Singh & Jaggi, 2020). Model Soups (Wortsman et al.,

2022) averages compatible fine-tuned models, while Fisher Merging (Matena & Raffel, 2022) and RegMean (Jin et al., 2023) use Fisher information and input activation statistics, respectively. Task Arithmetic (Ilharco et al., 2023) instead represents task-specific adaptation as task vectors and combines them through addition and scaling.

Pruning-based merging. TIES-Merging (Yadav et al., 2023) removes small-magnitude components and resolves sign conflicts, while DARE (Yu et al., 2024) randomly drops and rescales task-vector parameters. DELLA-Merging (Deep et al., 2024) further uses magnitude-aware stochastic pruning.

SVD-based merging. TSV-Merging (Gargiulo et al., 2025) reduces interference through singular-vector removal and whitening, while Isotropic Merging (Marczak et al., 2025) adjusts the singular-value spectrum over common and task-specific subspaces. Subspace Boosting (Skorobogat et al., 2026) addresses rank collapse by boosting small singular values.

Optimization-based merging. AdaMerging (Yang et al., 2024b) learns task- and layer-wise merging coefficients using unlabeled data, while AWD (Xiong et al., 2025) encourages orthogonality among task vectors. DOGE (Wei et al., 2025) formulates merging as constrained optimization, whereas WUDI-Merging (Cheng et al., 2025) constructs corrected merging directions from task vectors in a data-free manner.

Data-free activation-aware merging. ACTMat (Hameed et al., 2026) estimates activation covariances from task-vector Gram matrices for RegMean's activation-matching objective. GTA uses the same Gram covariances. Under Theorem 2's assumptions, converged PSI-Gates recover closed-form ACTMat, whereas finite steps from identity retain a preservation-oriented spectral bias. More generally, GTA explicitly writes each task contribution as $\boldsymbol{\tau}_t \boldsymbol{G}_t$, a form not tied to the PSI loss and potentially compatible with structured linear gates or other objectives.

3 METHODOLOGY

3.1 PRELIMINARIES

We first review task vectors and Task Arithmetic (Ilharco et al., 2023). For each task $t \in \mathcal{T}$, a task vector is defined as the parameter difference between the fine-tuned model and its shared base model: $\boldsymbol{\tau}_t = \boldsymbol{W}_t - \boldsymbol{W}_{\text{base}} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. Let $n = |\mathcal{T}|$ denote the number of tasks. Throughout this paper, task-vector operations are applied independently to each parameter matrix.

Task Arithmetic constructs a merged task vector by scaling the sum of individual task vectors:$\boldsymbol{\tau}_m^{(\text{TA})} = \alpha \sum_{t \in \mathcal{T}} \boldsymbol{\tau}_t$, where α is a scaling coefficient and $\boldsymbol{\tau}_m$ denotes the merged task vector.

3.2 GATED TASK ARITHMETIC WITH PSI-GATES

We consider an arbitrary linear layer l and omit the layer index l throughout this section for notational simplicity. Let $\mathcal{D}_t \subseteq \mathbb{R}^{d_{\text{in}}}$ denote the set of input representations associated with task t at this layer. For each task t, we consider a batch of input representations $\{\boldsymbol{h}_{t,j}\}_{j=1}^{N_t} \subseteq \mathcal{D}_t$ and define

$$\boldsymbol{H}_t = [\boldsymbol{h}_{t,1}, \ldots, \boldsymbol{h}_{t,N_t}] \in \mathbb{R}^{d_{\text{in}} \times N_t}. \tag{1}$$

We define *layer-level task interference* as the discrepancy between the outputs of the merged and task-specific weights on the same task-specific input representations:

$$\mathcal{I}(\boldsymbol{\tau}_m) = \sum_{t \in \mathcal{T}} \|(\boldsymbol{\tau}_m - \boldsymbol{\tau}_t)\boldsymbol{H}_t\|_F^2 . \tag{2}$$

For Task Arithmetic with $\alpha = 1$, Eq. 2 becomes $\mathcal{I}(\boldsymbol{\tau}_m^{(\text{TA})}) = \sum_{t \in \mathcal{T}} \left\| \sum_{i \in \mathcal{T},\, i \neq t} \boldsymbol{\tau}_i \boldsymbol{H}_t \right\|_F^2$. Thus, for the input representations of task t, interference arises from the task vectors associated with the other tasks.

To control this cross-task interaction, we introduce **Gated Task Arithmetic (GTA)**. For each task vector $\boldsymbol{\tau}_t$, GTA introduces an input-side linear gate $\boldsymbol{G}_t \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ and defines the gated task vector as $\widetilde{\boldsymbol{\tau}}_t = \boldsymbol{\tau}_t \boldsymbol{G}_t$. The resulting merged task vector is $\boldsymbol{\tau}_m^{(\text{GTA})} = \sum_{t \in \mathcal{T}} \boldsymbol{\tau}_t \boldsymbol{G}_t$.

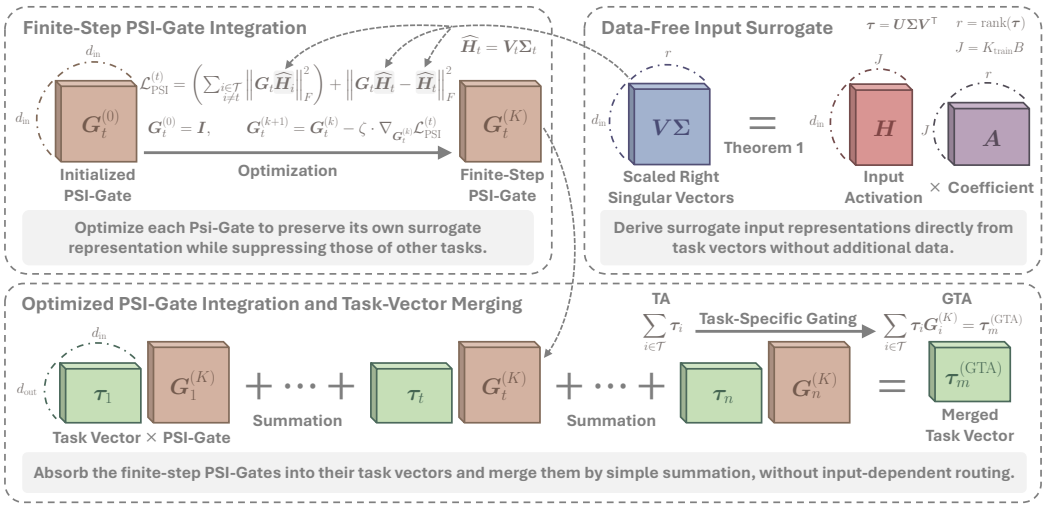


Figure 2: Overview of **GTA**. For each task vector, GTA constructs a data-free surrogate input representation from its scaled right singular vectors and initializes the corresponding task-preserving and interference-suppressing gate (PSI-Gate) with the identity matrix. Each PSI-Gate is optimized for K finite steps to preserve its own surrogate representation while suppressing those of the other tasks. The resulting finite-step PSI-Gates are absorbed into their task vectors by right multiplication, and the gated task vectors are merged by simple summation without input-dependent routing.

We design $\boldsymbol{G}_t$ as a **task-preserving and interference-suppressing gate (PSI-Gate)**. Ideally, the PSI-Gate for task t should satisfy

$$\boldsymbol{G}_t \boldsymbol{H}_i = \begin{cases} \boldsymbol{H}_i, & i = t, \\ \boldsymbol{0}, & i \neq t, \end{cases} \tag{3}$$

corresponding to task preservation and interference suppression, respectively.

Proposition 1 (Interference Elimination under Ideal PSI-Gates)**.** *If all PSI-Gates satisfy Eq. 3, the layer-level task interference of GTA is zero:* $\mathcal{I}(\boldsymbol{\tau}_m^{(\mathrm{GTA})}) = 0$.

This follows because, for each task t, $\boldsymbol{G}_t \boldsymbol{H}_t = \boldsymbol{H}_t$ and $\boldsymbol{G}_i \boldsymbol{H}_t = \boldsymbol{0}$ for all $i \neq t$.

3.3 Data-Free Surrogate Input Representations

The ideal PSI-Gate conditions require task-specific input representations $\boldsymbol{H}_t$, which are unavailable in data-free model merging. We therefore derive input-side information directly from the task vectors. The following theorem establishes a connection between a task vector and the input representations observed during fine-tuning.

Theorem 1 (Input-Activation Representation of Task Vectors)**.** *Under the SGD assumptions stated in Appendix A.1, let* $\boldsymbol{\tau} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ *be the compact SVD of a task vector with rank* r*, and let* $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_J]$ *contain the input representations observed during fine-tuning. Then,*

$$\boldsymbol{V}\boldsymbol{\Sigma} = \boldsymbol{H}\boldsymbol{A}, \tag{4}$$

where $[\boldsymbol{A}]_{j,i} = \lambda_j(\boldsymbol{g}_j^\top \boldsymbol{u}_i)$*,* $\boldsymbol{g}_j$ *is the corresponding output gradient, and* λ_j *is determined by the learning rate and batch size.*

The complete statement of Theorem 1, including its assumptions and proof, is provided in Appendix A.1. The theorem shows that $\boldsymbol{V}\boldsymbol{\Sigma}$ consists of gradient-weighted linear combinations of the input representations observed during fine-tuning. This leads to the following corollary.

Corollary 1 (Task-Vector-Derived Surrogate Input Representation)**.** *For task* t*, define* $\widehat{\boldsymbol{H}}_t := \boldsymbol{V}_t\boldsymbol{\Sigma}_t$*. Under the assumptions of Theorem 1,*

$$\widehat{\boldsymbol{H}}_t = \boldsymbol{H}_t \boldsymbol{A}_t, \tag{5}$$

and hence $\widehat{\boldsymbol{H}}_t$ *lies in the span of the input representations observed during fine-tuning on task* t*.*

Based on Corollary 1, we use $\widehat{\boldsymbol{H}}_t$ as a task-vector-derived surrogate for the input-side information of task t. Note that $\widehat{\boldsymbol{H}}_t$ does not reconstruct the original input representations $\boldsymbol{H}_t$; instead, it represents particular gradient-weighted linear combinations of them.

3.4 PSI Surrogate Loss

Using the surrogate input representations $\{\widehat{\boldsymbol{H}}_i\}_{i\in\mathcal{T}}$, we learn each PSI-Gate $\boldsymbol{G}_t$ with two complementary objectives: task preservation and interference suppression.

Task preservation. To preserve the input information associated with task t, we define the task-preservation loss as $\mathcal{L}_{\text{pre}}^{(t)} = \|\boldsymbol{G}_t\widehat{\boldsymbol{H}}_t - \widehat{\boldsymbol{H}}_t\|_F^2$. This loss encourages the PSI-Gate to retain the task-specific input information encoded in $\widehat{\boldsymbol{H}}_t$.

Interference suppression. To suppress input information associated with other tasks, we define the interference-suppression loss as $\mathcal{L}_{\text{sup}}^{(t)} = \sum_{i\in\mathcal{T},\, i\neq t} \|\boldsymbol{G}_t\widehat{\boldsymbol{H}}_i\|_F^2$. This loss encourages the PSI-Gate to suppress input information encoded in the surrogate representations of the other tasks.

PSI surrogate loss. Combining the two objectives, we define the **task-preserving and interference-suppressing surrogate loss (PSI surrogate loss)** as

$$\mathcal{L}_{\text{PSI}}^{(t)} = \mathcal{L}_{\text{pre}}^{(t)} + \mathcal{L}_{\text{sup}}^{(t)}. \tag{6}$$

The PSI surrogate loss encourages the PSI-Gate to approximate the ideal conditions in Eq. 3 using only information derived from the task vectors, without access to the original training data.

3.5 Gate Optimization and Gated Merging

We initialize each PSI-Gate with the identity matrix, $\boldsymbol{G}_t^{(0)} = \boldsymbol{I}$, and optimize only the gate with Adam while keeping the task vectors and surrogate input representations fixed. To characterize finite-step optimization analytically, we consider gradient descent, under which the gate is updated as $\boldsymbol{G}_t^{(k+1)} = \boldsymbol{G}_t^{(k)} - \zeta\nabla_{\boldsymbol{G}_t^{(k)}}\mathcal{L}_{\text{PSI}}^{(t)}$, where ζ is the learning rate.

Although the PSI surrogate loss admits a closed-form solution, GTA performs a finite number of optimization steps from the identity initialization. The following theorem characterizes the resulting optimization bias in this analytically tractable gradient-descent setting. The complete analysis and proof are provided in Appendix A.2.

Theorem 2 (Identity-Preserving Bias of Finite-Step Optimization)**.** *Let $\boldsymbol{G}_t^*$ denote the optimal PSI-Gate and define $\boldsymbol{S} = \sum_{i\in\mathcal{T}} \widehat{\boldsymbol{H}}_i\widehat{\boldsymbol{H}}_i^\top$. Under the conditions stated in Appendix A.2, gradient descent initialized with $\boldsymbol{G}_t^{(0)} = \boldsymbol{I}$ yields*

$$\boldsymbol{G}_t^{(K)} = \boldsymbol{G}_t^* + (\boldsymbol{I} - \boldsymbol{G}_t^*)(\boldsymbol{I} - 2\zeta\boldsymbol{S})^K. \tag{7}$$

Hence, finite-step optimization retains an identity-preserving residual whose decay is determined by the spectrum of $\boldsymbol{S}$.

Theorem 2 shows that finite-step optimization biases smaller-eigenvalue directions of $\boldsymbol{S}$ more strongly toward identity. When task-vector Gram matrices are used as the covariance estimates of ACTMat (Hameed et al., 2026), the limiting merge at $K \to \infty$ coincides with closed-form ACTMat. Thus, GTA differs through its finite-step identity-preserving residual rather than its limiting quadratic solution.

After K optimization steps, GTA constructs the merged task vector as

$$\boldsymbol{\tau}_m^{(\text{GTA})} = \sum_{t\in\mathcal{T}} \boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}. \tag{8}$$

The merged task vector is then added to the base model parameters to obtain the final merged model. Since each PSI-Gate is absorbed into its corresponding task vector through matrix multiplication, GTA requires no input-dependent routing or additional modules at inference time.

4 EXPERIMENTS

4.1 SETTINGS

Models & Datasets We evaluate GTA across diverse task types, model architectures, and model sizes using models fine-tuned on vision, natural language classification, and reasoning tasks. We first use task-specific fine-tuned models provided by *FusionBench* (Tang et al., 2025) to evaluate the merging of ViT (Dosovitskiy et al., 2021) models across 20 vision tasks. We also evaluate LLaMA (Grattafiori et al., 2024) models fine-tuned on 8 natural language classification tasks to examine merging performance across different model sizes, including LLaMA-3.2-1B, LLaMA-3.2-3B, and LLaMA-3.1-8B. To extend our evaluation beyond classification to natural language reasoning, we merge Qwen2-7B (Yang et al., 2024a) models fine-tuned on 5 reasoning datasets with chain-of-thought (CoT) reasoning. Together, these settings evaluate GTA across a range of model merging scenarios rather than a single task type or model family. Details of the models and datasets are provided in Appendix C.

Hyperparameters We configure the main hyperparameters of the comparison methods and GTA as follows. For Task Arithmetic (TA) (Ilharco et al., 2023), we set the merging coefficient to $\alpha = 1/n$, where n is the number of tasks being merged. This experimental setting differs from the reference setting $\alpha = 1$ used in our interference analysis. For TIES (Yadav et al., 2023), we retain a fraction of 0.2 of the task-vector parameters after pruning. For Task Singular Vectors (TSV) (Gargiulo et al., 2025), we set the rank used for merging to r/n, while for Subspace Boosting (Skorobogat et al., 2026), we use $\beta = 0.0$ to control singular-value flattening. For WUDI (Cheng et al., 2025), we follow the hyperparameter settings reported in the original paper. For **GTA**, we initialize each PSI-Gate with the identity matrix and optimize only the gates while keeping the task vectors and their surrogate input representations fixed. Specifically, we minimize the PSI surrogate loss in Eq. 6 using Adam (Kingma & Ba, 2017) for $K = 300$ gate updates with a learning rate of $\zeta = 10^{-3}$.

Evaluation Metrics We evaluate task performance using accuracy (ACC), with prediction procedures tailored to each task type. For the 20 vision tasks, we follow the task-specific evaluation procedures provided by FusionBench. For the 8 natural language classification tasks, we use the logit-based evaluation procedure in *LM Eval Harness* (Gao et al., 2024) to obtain model predictions and compute accuracy against the ground-truth labels. For the 5 natural language reasoning tasks, we extract the final answer from the generated text using the designated answer markers and evaluate it using exact match (EM) against the reference answer. Thus, classification evaluation measures the correctness of predicted labels, whereas reasoning evaluation measures the correctness of generated final answers.

We additionally introduce **Task Recovery Rate (TRR)** as a complementary metric to quantify how much of the performance gain from fine-tuning is preserved after merging. Absolute accuracy alone does not distinguish the base model's existing capabilities from the additional gains obtained through fine-tuning. In particular, when the base model already achieves high accuracy on a task, a merged model may retain high accuracy even if much of the fine-tuning gain is lost. TRR therefore normalizes the improvement of the merged model over the base model by the corresponding improvement of the task-specific fine-tuned model. Let $\mathrm{ACC}_t(\boldsymbol{W})$ denote the accuracy of a model with parameters $\boldsymbol{W}$ on task t, and let $\boldsymbol{W}_{\mathrm{base}}$, $\boldsymbol{W}_t$, and $\boldsymbol{W}_m$ denote the parameters of the base model, the model fine-tuned on task t, and the merged model, respectively. We define TRR as

$$\mathrm{TRR}_t(\boldsymbol{W}_m) = \frac{\mathrm{ACC}_t(\boldsymbol{W}_m) - \mathrm{ACC}_t(\boldsymbol{W}_{\mathrm{base}})}{\mathrm{ACC}_t(\boldsymbol{W}_t) - \mathrm{ACC}_t(\boldsymbol{W}_{\mathrm{base}})}. \tag{9}$$

When the fine-tuned model outperforms the base model, TRR represents the fraction of the fine-tuning gain recovered by the merged model. A TRR of 0 indicates performance equal to that of the base model, whereas a TRR of 1 indicates performance equal to that of the task-specific fine-tuned model. TRR therefore provides a relative measure for comparing merging performance across tasks with different base accuracies and fine-tuning gains. However, when the fine-tuning gain is small, even a minor difference in accuracy can cause a large change in TRR. We therefore report and interpret TRR together with absolute accuracy. TRR is undefined when the fine-tuned and base models have equal accuracy, since its denominator is zero.

4.2 RESULTS

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 Tasks	14 Tasks	20 Tasks	8 Tasks	14 Tasks	20 Tasks	8 Tasks	14 Tasks	20 Tasks
Non-Merged									
Base	48.08	56.94	55.65	55.36	61.97	59.85	64.93	69.10	65.61
Fine-Tuned	90.38	90.38	89.77	92.38	91.33	91.62	94.30	93.37	93.52
Merged									
TA	66.32 (43.1%)	65.36 (25.2%)	61.10 (16.0%)	72.33 (45.8%)	69.75 (26.5%)	64.81 (15.6%)	79.87 (50.9%)	77.53 (34.7%)	71.15 (19.8%)
TIES	69.59 (50.9%)	62.33 (16.1%)	54.42 (-3.6%)	78.24 (61.8%)	67.31 (18.2%)	59.87 (0.1%)	82.49 (59.8%)	74.84 (23.6%)	65.08 (-1.9%)
TSV	83.07 (82.7%)	78.79 (65.4%)	73.22 (51.5%)	87.10 (85.7%)	82.13 (68.7%)	78.31 (58.1%)	90.57 (87.3%)	88.21 (78.7%)	83.19 (63.0%)
Subspace Boosting	84.04 (85.0%)	77.32 (61.0%)	70.19 (42.6%)	89.03 (91.0%)	81.75 (67.4%)	73.55 (43.1%)	91.57 (90.7%)	88.33 (79.2%)	80.71 (54.1%)
WUDI	84.60 (86.3%)	79.49 (67.4%)	71.02 (45.1%)	89.15 (91.3%)	83.77 (74.3%)	76.03 (50.9%)	92.24 (93.0%)	89.50 (84.1%)	81.92 (58.4%)
ACTMat	82.83 (82.2%)	76.96 (59.9%)	72.23 (48.6%)	88.83 (90.4%)	81.24 (65.6%)	78.23 (57.9%)	92.10 (92.5%)	89.33 (83.4%)	85.94 (72.8%)
GTA	**86.51 (90.8%)**	**81.60 (73.8%)**	**77.39 (63.7%)**	**90.28 (94.3%)**	**85.22 (79.2%)**	**83.38 (74.1%)**	**93.09 (95.9%)**	**91.13 (90.8%)**	**89.37 (85.1%)**

Table 1: Model merging performance on vision tasks using ViT-B/32, ViT-B/16, and ViT-L/14. Each column reports the average accuracy (%) over the corresponding set of 8, 14, or 20 tasks. Gray values in parentheses indicate the Task Recovery Rate (TRR, %), computed using the average accuracies of the merged, base, and task-specific fine-tuned models. *Base* and *Fine-Tuned* provide non-merged reference results.

Method	GSM8K	SQA	OBQA	E-SNLI	ECQA	Avg.
Non-Merged						
Base	18.20	62.30	78.60	71.97	67.59	59.73
Fine-Tuned	76.04	74.24	88.60	90.60	79.54	81.80
Merged						
TA	75.82 (99.6%)	72.34 (84.1%)	82.60 (40.0%)	89.99 (96.7%)	76.94 (78.2%)	79.54 (89.7%)
TIES	77.10 (101.8%)	70.89 (72.0%)	81.00 (24.0%)	90.80 (101.0%)	78.40 (90.5%)	79.64 (90.2%)
TSV	74.83 (97.9%)	72.78 (87.8%)	83.40 (48.0%)	90.78 (100.9%)	78.90 (94.7%)	80.14 (92.4%)
Subspace Boosting	74.45 (97.2%)	75.25 (108.5%)	83.60 (50.0%)	90.39 (98.9%)	78.53 (91.6%)	80.45 (93.8%)
WUDI	75.06 (98.3%)	73.51 (93.9%)	86.00 (74.0%)	90.83 (101.2%)	77.62 (84.0%)	80.60 (94.6%)
ACTMat	75.28 (98.7%)	73.65 (95.1%)	84.20 (56.0%)	90.70 (100.5%)	78.58 (92.0%)	80.48 (94.0%)
GTA	**75.21 (98.6%)**	**75.55 (111.0%)**	**86.00 (74.0%)**	**90.54 (99.7%)**	**78.76 (93.5%)**	**81.21 (97.3%)**

Table 2: Model merging performance on five natural language reasoning tasks using Qwen2-7B. Each task column reports exact match (EM, %) for the final answers extracted from the generated text, while *Avg.* reports the average EM over all five tasks. Gray values in parentheses indicate the Task Recovery Rate (TRR, %). For *Avg.*, TRR is computed using the average EM scores of the merged, base, and task-specific fine-tuned models. *Base* and *Fine-Tuned* are shown as non-merged references.

Merging Results across Vision and Language Tasks Table 1 shows that GTA achieves the highest average accuracy and corresponding TRR among the compared merging methods across all evaluated ViT architectures and task-count settings. In the 20-task setting, GTA outperforms the strongest baseline by 4.16, 5.15, and 3.43 percentage points in average accuracy on ViT-B/32, ViT-B/16, and ViT-L/14, respectively, with corresponding TRR improvements of 15.1, 16.2, and 12.3

percentage points. These improvements indicate that the benefit of GTA extends beyond settings with a small number of task-specific models. The higher TRR further shows that GTA preserves a larger fraction of the performance gains obtained through fine-tuning, rather than merely retaining the capabilities of the base model.

Table 2 shows that GTA also performs strongly on generative language reasoning tasks with Qwen2-7B. GTA achieves the highest average exact-match score of 81.21% and the highest average TRR of 97.3%, outperforming the strongest baseline by 0.61 and 2.7 percentage points, respectively. Across the three LLaMA model sizes, GTA outperforms ACTMat by 1.45 percentage points on LLaMA-3.2-1B, performs comparably on LLaMA-3.2-3B (78.07 vs. 78.09), and outperforms it by 0.39 percentage points on LLaMA-3.1-8B. Detailed task-level results for all three LLaMA scales are provided in Appendix D. Together, these results demonstrate that GTA remains effective across model families, model scales, task types, and evaluation protocols.

These results are consistent with the finite-step perspective of Theorem 2: under its gradient-descent assumptions, optimization from the identity retains an identity-preserving residual rather than immediately reaching the quadratic optimum, but the theorem does not predict downstream accuracy.

Interference Reduction on Real Input Activations We evaluate whether PSI-Gates learned from the data-free surrogate $\widehat{\boldsymbol{H}}_t = \boldsymbol{V}_t \boldsymbol{\Sigma}_t$, derived solely from the task vector, reduce interference on real input activations. Following Eq. 2, we measure interference before and after gating using the same fixed input activation matrix $\boldsymbol{H}_t$ for each task t. We use Task Arithmetic with $\alpha = 1$ as the ungated reference and compute the following quantities for each layer:

$$\begin{aligned} \mathcal{I}^{(\mathrm{TA})} &= \sum_{t \in \mathcal{T}} \left\| \left(\sum_{i \in \mathcal{T}} \boldsymbol{\tau}_i - \boldsymbol{\tau}_t \right) \boldsymbol{H}_t \right\|_F^2, \\ \mathcal{I}^{(\mathrm{GTA})} &= \sum_{t \in \mathcal{T}} \left\| \left(\sum_{i \in \mathcal{T}} \boldsymbol{\tau}_i \boldsymbol{G}_i - \boldsymbol{\tau}_t \right) \boldsymbol{H}_t \right\|_F^2. \end{aligned} \tag{10}$$

Here, $\mathcal{I}^{(\mathrm{TA})}$ and $\mathcal{I}^{(\mathrm{GTA})}$ measure the layer-level task interference before and after gating, respectively.

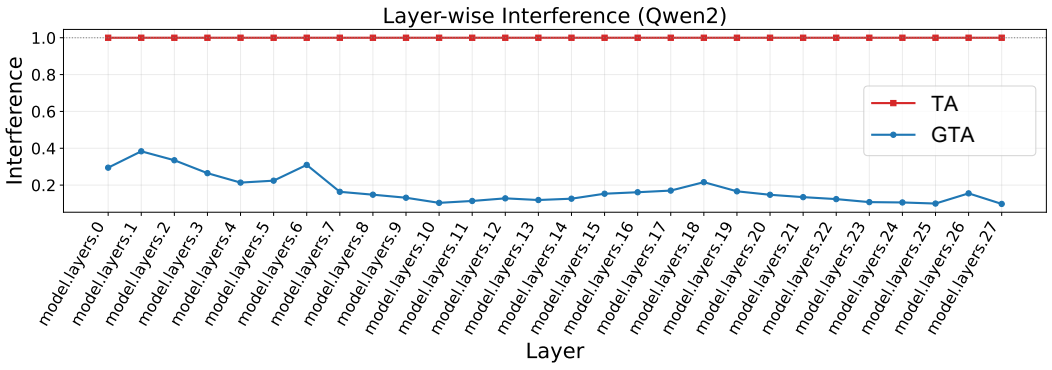


Figure 3: Layer-level task interference on real input activations for Qwen2-7B when merging five natural language reasoning tasks, measured using 100 samples per task with a fixed random seed of 42. Interference is normalized by the ungated Task Arithmetic reference: "TA" corresponds to $\mathcal{I}^{(\mathrm{TA})}/\mathcal{I}^{(\mathrm{TA})} = 1$, while "GTA" corresponds to $\mathcal{I}^{(\mathrm{GTA})}/\mathcal{I}^{(\mathrm{TA})}$. Values are averaged within each layer for visualization, and values below 1 indicate reduced interference after gating.

Figure 3 summarizes the measured layer-level interference for Qwen2-7B on natural language reasoning tasks. We normalize the interference by the ungated Task Arithmetic reference, such that a ratio $\mathcal{I}^{(\mathrm{GTA})}/\mathcal{I}^{(\mathrm{TA})} < 1$ indicates reduced interference after gating. The reduction observed on real input activations supports the use of $\boldsymbol{V}_t \boldsymbol{\Sigma}_t$ as a task-vector-derived surrogate for learning PSI-Gates that suppress cross-task interference. Together with the vision and classification results, this reasoning-task analysis is consistent with the intended role of PSI-Gates in suppressing cross-task

interference while preserving task-specific information across different task types. Results for other models are provided in Appendix E.

Preservation of Individual Task-Vector Performance. The preceding analysis shows that GTA reduces cross-task interference on real input activations. We next examine whether the PSI-Gates preserve the task-specific capabilities of the individual task vectors. For each task t, we independently apply the original task vector $\boldsymbol{\tau}_t$ and its gated counterpart $\boldsymbol{\tau}_t \boldsymbol{G}_t^{(K)}$ to the base model and evaluate both on task t.

Method	GSM8K	SQA	OBQA	E-SNLI	ECQA	Avg.
Original						
$\boldsymbol{W}_{\text{base}}$	18.20	62.30	78.60	71.97	67.59	59.73
$\boldsymbol{\tau}_t$	76.04	74.24	88.60	90.60	79.54	81.80
Gated						
$\boldsymbol{\tau}_t \boldsymbol{G}_t$	76.12 (100.1%) +0.08	75.69 (112.2%) +1.46	86.60 (80.0%) -2.00	90.81 (101.1%) +0.20	78.99 (95.4%) -0.55	81.64 (99.3%) -0.16

Table 3: Individual task-vector performance before and after PSI-Gating on Qwen2-7B. For each task t, $\boldsymbol{\tau}_t$ denotes the original task vector, while $\boldsymbol{\tau}_t \boldsymbol{G}_t^{(K)}$ denotes its gated counterpart. Each entry reports exact match (EM, %) on task t when the corresponding task vector is applied individually to the base model. Gray values in parentheses indicate the Task Recovery Rate (TRR, %), computed using $\boldsymbol{W}_{\text{base}}$, $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t$, and $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t \boldsymbol{G}_t^{(K)}$, with the original task vector serving as the task-specific reference. The bottom row reports the EM difference between $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t \boldsymbol{G}_t^{(K)}$ and $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t$.

As shown in Table 3, gating largely preserves individual task-vector performance on the reasoning tasks: the average EM decreases by only 0.16 percentage points, from 81.80% to 81.64%, while the average TRR remains 99.3%. Improvements on several tasks further indicate that the PSI-Gates do not merely weaken the task vectors. Together with the reduced interference, these results show that PSI-Gates suppress cross-task interference while preserving task-specific capabilities. Results for the other models are provided in Appendix F.

Appendix H further isolates the role of identity initialization under a fixed finite-step budget. Although Identity yields higher suppression and total PSI losses than Constant and Random, it maintains a lower preservation loss and achieves the best downstream accuracy (81.78% versus 81.34% and 81.27%). Together with Theorem 2 and the limiting ACTMat comparison, these results support finite-step GTA as a preservation-oriented design: a lower surrogate loss does not necessarily yield better merging performance.

5 Conclusion

We introduced **Gated Task Arithmetic (GTA)**, a data-free method that learns PSI-Gates from task-vector-derived surrogate input representations and absorbs them into the task vectors, requiring neither additional data nor input-dependent routing at inference time. We further showed that finite-step optimization from identity induces a preservation-oriented spectral bias, while its $K \to \infty$ limit recovers closed-form ACTMat under task-vector Gram covariance estimates.

Across vision and language models, GTA achieved strong merging performance and task recovery. Real-activation and individual-vector analyses linked these gains to reduced interference and preserved task-specific capabilities. Notably, Identity achieved the best downstream performance despite a higher total PSI loss, showing that fully minimizing the surrogate objective need not produce the best merged model. Although separate PSI-Gates add merging-time optimization cost, these results establish task-specific input gating and finite-step optimization as effective design principles for data-free model merging.

AI USE STATEMENT

Generative AI tools were used to assist with the writing and editing of the manuscript, including improving clarity, readability, and overall language quality. They were also used to help refine the presentation of technical descriptions and explanations.

REFERENCES

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. Explanations for CommonsenseQA: New Dataset and Models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3050–3065, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.238. URL https://aclanthology.org/2021.acl-long.238/.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022. URL https://arxiv.org/abs/2108.07258.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations, 2018. URL https://arxiv.org/abs/1812.01193.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. doi: 10.1109/JPROC.2017.2675998.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors, 2025. URL https://arxiv.org/abs/2503.08099.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *ArXiv*, abs/1812.01718, 2018. URL https://arxiv.org/abs/1812.01718.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/coates11a.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters, 2017. URL https://arxiv.org/abs/1702.05373.

Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. Della-merging: Reducing interference in model merging through magnitude-based sampling, 2024. URL https://arxiv.org/abs/2406.11617.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://aclanthology.org/I05-5002/.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

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

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl_a_00370. URL https://aclanthology.org/2021.tacl-1.21/.

Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in representation learning: A report on three machine learning contests, 2013. URL https://arxiv.org/abs/1307.0414.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco

Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L.

Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Marawan Gamal Abdel Hameed, Derek Tam, Pascal Jr Tikeng Notsawo, Colin Raffel, and Guillaume Rabusseau. Model merging via data-free covariance estimation, 2026. URL https://arxiv.org/abs/2604.01329.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 204–207, 2018. doi: 10.1109/IGARSS.2018.8519248.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

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

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, June 2013.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=RBZpAa27ls.

Michael S Matena and Colin Raffel. Merging models with fisher-weighted averaging. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17703–17716. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/70c26937fbf3d4600b69a129031b66ec-Paper-Conference.pdf.

Sachit Menon and Carl Vondrick. Visual classification via description from large language models, 2022. URL https://arxiv.org/abs/2210.07183.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092.

Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1267–1273, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1128. URL https://aclanthology.org/N19-1128/.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019. URL https://arxiv.org/abs/1907.10641.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22045–22055. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/fb2697869f56484404c8ceee2985b01d-Paper.pdf.

Ronald Skorobogat, Karsten Roth, and Iuliana Georgescu. Subspace-boosted model merging, 2026. URL https://openreview.net/forum?id=0Rn9oMU6g9.

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2012.02.016. URL https://www.sciencedirect.com/science/article/pii/S0893608012000457. Selected Papers from IJCNN 2011.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and

Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Anke Tang, Li Shen, Yong Luo, Enneng Yang, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. Fusionbench: A comprehensive benchmark of deep model fusion. *Journal of Machine Learning Research*, 2025.

Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology, 2018. URL https://arxiv.org/abs/1806.03962.

Yongxian Wei, Anke Tang, Li Shen, Zixuan Hu, Chun Yuan, and Xiaochun Cao. Modeling multi-task model merging as adaptive projective gradient descent. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EqoKRSR5Pa.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101/.

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 Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL https://arxiv.org/abs/1708.07747.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010. doi: 10.1109/CVPR.2010.5539970.

Feng Xiong, Runxi Cheng, Wang Chen, Zhanqiu Zhang, Yiwen Guo, Chun Yuan, and Ruifeng Xu. Multi-task model merging via adaptive weight disentanglement, 2025. URL https://arxiv.org/abs/2411.18729.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024a. URL https://arxiv.org/abs/2407.10671.

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

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, 2024. URL https://arxiv.org/abs/2311.03099.

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

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2016. URL https://arxiv.org/abs/1509.01626.

A Proof of Theorems

A.1 Proof of Theorem 1: Analysis of Input-Activation Representation

Theorem 1 (Input-activation representation of task vectors). Consider a linear layer $\boldsymbol{z} = \boldsymbol{W}\boldsymbol{h}$, and let $\boldsymbol{W}_{\text{base}}$ and $\boldsymbol{W}_{\text{task}}$ denote its initial and fine-tuned weights, respectively. Define the task vector as

$$\boldsymbol{\tau} = \boldsymbol{W}_{\text{task}} - \boldsymbol{W}_{\text{base}} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}. \tag{11}$$

For each sample, assume that the input $\boldsymbol{h}$ does not depend on the weights $\boldsymbol{W}$ of this layer and that the loss depends on $\boldsymbol{W}$ only through $\boldsymbol{z} = \boldsymbol{W}\boldsymbol{h}$. Furthermore, assume that $\boldsymbol{W}$ is updated directly by SGD on the mini-batch mean loss, without momentum, weight decay, or additional gradient transformations.

Let $k = 0, \ldots, K-1$ index the training steps, let B denote the batch size, and let η_k denote the learning rate at step k. For sample b at step k, let $\boldsymbol{h}_{k,b} \in \mathbb{R}^{d_{\text{in}}}$ denote the input activation and $\boldsymbol{g}_{k,b} = \partial \ell_{k,b} / \partial \boldsymbol{z}_{k,b} \in \mathbb{R}^{d_{\text{out}}}$ the gradient with respect to the layer output. Set $j = kB + b$ and $J = KB$, and write $\boldsymbol{h}_j = \boldsymbol{h}_{k,b}$, $\boldsymbol{g}_j = \boldsymbol{g}_{k,b}$, and $\lambda_j = -\eta_k / B$.

Let $\boldsymbol{\tau} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ be the compact singular value decomposition of $\boldsymbol{\tau}$, with $r = \operatorname{rank}(\boldsymbol{\tau})$, and let $\boldsymbol{u}_i$ denote the ith column of $\boldsymbol{U}$. Define

$$\begin{aligned} \boldsymbol{H} &= [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_J] \in \mathbb{R}^{d_{\text{in}} \times J}, \qquad \boldsymbol{A} \in \mathbb{R}^{J \times r}, \\ [\boldsymbol{A}]_{j,i} &= \lambda_j (\boldsymbol{g}_j^\top \boldsymbol{u}_i), \qquad j = 1, \ldots, J, \quad i = 1, \ldots, r. \end{aligned} \tag{12}$$

Then,

$$\boldsymbol{V}\boldsymbol{\Sigma} = \boldsymbol{H}\boldsymbol{A}. \tag{13}$$

Proof. We first express the task vector as a sum of outer products obtained from the SGD updates and then use the compact singular value decomposition to derive Eq. 13.

1. Gradient of the linear layer. Consider a single sample and omit the step and sample indices. Let $\boldsymbol{h}$ denote the input activation, $\boldsymbol{z} = \boldsymbol{W}\boldsymbol{h}$ the layer output, and ℓ the loss. Define $\boldsymbol{g} = \partial \ell / \partial \boldsymbol{z}$. The pth output component is

$$z_p = \sum_{q=1}^{d_{\text{in}}} [\boldsymbol{W}]_{pq} h_q. \tag{14}$$

By assumption, $\boldsymbol{h}$ does not depend on $\boldsymbol{W}$, and the loss depends on $\boldsymbol{W}$ only through $\boldsymbol{z}$. The chain rule therefore gives

$$\frac{\partial \ell}{\partial [\boldsymbol{W}]_{pq}} = \sum_{s=1}^{d_{\text{out}}} \frac{\partial \ell}{\partial z_s} \frac{\partial z_s}{\partial [\boldsymbol{W}]_{pq}} = g_p h_q. \tag{15}$$

Hence,

$$\nabla_{\boldsymbol{W}} \ell = \boldsymbol{g}\boldsymbol{h}^\top. \tag{16}$$

2. Accumulating the SGD updates. Let $\boldsymbol{W}^{(k)}$ denote the weights at step k and define the mini-batch mean loss as

$$\mathcal{L}^{(k)} = \frac{1}{B} \sum_{b=1}^{B} \ell_{k,b}. \tag{17}$$

The corresponding gradient is

$$\nabla_{\boldsymbol{W}} \mathcal{L}^{(k)} = \frac{1}{B} \sum_{b=1}^{B} \boldsymbol{g}_{k,b} \boldsymbol{h}_{k,b}^\top. \tag{18}$$

All activations and gradients are evaluated at their corresponding training steps. The SGD update is

$$\boldsymbol{W}^{(k+1)} - \boldsymbol{W}^{(k)} = -\eta_k \nabla_{\boldsymbol{W}} \mathcal{L}^{(k)}. \tag{19}$$

Since $\boldsymbol{W}^{(0)} = \boldsymbol{W}_{\text{base}}$ and $\boldsymbol{W}^{(K)} = \boldsymbol{W}_{\text{task}}$, summing the K updates gives

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{W}^{(K)} - \boldsymbol{W}^{(0)} \\ &= \sum_{k=0}^{K-1} \left(\boldsymbol{W}^{(k+1)} - \boldsymbol{W}^{(k)} \right) \\ &= -\sum_{k=0}^{K-1} \sum_{b=1}^{B} \frac{\eta_k}{B} \boldsymbol{g}_{k,b} \boldsymbol{h}_{k,b}^{\top} \\ &= \sum_{j=1}^{J} \lambda_j \boldsymbol{g}_j \boldsymbol{h}_j^{\top}.\end{aligned} \tag{20}$$

This identity holds exactly under the stated SGD assumptions. The columns of $\boldsymbol{H}$ therefore correspond to input activations observed across training steps, rather than activations collected from a single fixed model.

3. Input-activation representation via SVD. From the compact SVD $\boldsymbol{\tau} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}$ and $\boldsymbol{U}^{\top}\boldsymbol{U} = \boldsymbol{I}_r$, we have

$$\boldsymbol{\tau}^{\top}\boldsymbol{U} = \boldsymbol{V}\boldsymbol{\Sigma}. \tag{21}$$

On the other hand, Eq. 20 gives

$$\boldsymbol{\tau}^{\top}\boldsymbol{U} = \sum_{j=1}^{J} \lambda_j \boldsymbol{h}_j \boldsymbol{g}_j^{\top}\boldsymbol{U}. \tag{22}$$

Its ith column is

$$\begin{aligned}\boldsymbol{\tau}^{\top}\boldsymbol{u}_i &= \sum_{j=1}^{J} \lambda_j (\boldsymbol{g}_j^{\top}\boldsymbol{u}_i)\boldsymbol{h}_j \\ &= \sum_{j=1}^{J} \boldsymbol{h}_j [\boldsymbol{A}]_{j,i} = \sigma_i \boldsymbol{v}_i.\end{aligned} \tag{23}$$

Since $\sum_{j=1}^{J} \boldsymbol{h}_j [\boldsymbol{A}]_{j,i}$ is the ith column of $\boldsymbol{H}\boldsymbol{A}$, collecting all columns gives

$$\boldsymbol{V}\boldsymbol{\Sigma} = \boldsymbol{\tau}^{\top}\boldsymbol{U} = \boldsymbol{H}\boldsymbol{A}, \tag{24}$$

which proves Eq. 13. □

A.1.1 Extension beyond the SGD Assumptions

Theorem 1 holds exactly under the SGD assumptions stated above. When the actual fine-tuning procedure deviates from these assumptions, the task vector may differ from the outer-product representation in Eq. 20. We characterize this discrepancy by defining

$$\boldsymbol{E} := \boldsymbol{\tau} - \sum_{j=1}^{J} \lambda_j \boldsymbol{g}_j \boldsymbol{h}_j^{\top} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}. \tag{25}$$

Here, $\boldsymbol{\tau}$ denotes the actual weight difference obtained after fine-tuning, with compact SVD $\boldsymbol{\tau} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}$. Using the same definitions of $\boldsymbol{H}$ and $\boldsymbol{A}$ as in Theorem 1, we obtain the exact identity

$$\begin{aligned}\boldsymbol{V}\boldsymbol{\Sigma} &= \boldsymbol{\tau}^{\top}\boldsymbol{U} \\ &= \left(\sum_{j=1}^{J} \lambda_j \boldsymbol{g}_j \boldsymbol{h}_j^{\top} + \boldsymbol{E} \right)^{\top} \boldsymbol{U} \\ &= \boldsymbol{H}\boldsymbol{A} + \boldsymbol{E}^{\top}\boldsymbol{U}.\end{aligned} \tag{26}$$

Therefore, when the projected residual $\boldsymbol{E}^{\top}\boldsymbol{U}$ is sufficiently small,

$$\boldsymbol{V}\boldsymbol{\Sigma} \approx \boldsymbol{H}\boldsymbol{A}. \tag{27}$$

This approximation is conditional on the actual optimization dynamics and is not guaranteed beyond the SGD assumptions of Theorem 1. When the projected residual is small, $\boldsymbol{V}\boldsymbol{\Sigma}$ can be interpreted as an approximate representation of gradient- and learning-rate-weighted combinations of the input activations observed during fine-tuning. It does not reconstruct the full activation matrix $\boldsymbol{H}$, nor does it imply that optimization problems defined using $\boldsymbol{V}\boldsymbol{\Sigma}$ and $\boldsymbol{H}$ are equivalent. We therefore use $\boldsymbol{V}\boldsymbol{\Sigma}$ only as a task-vector-derived surrogate input representation in the main method.

A.2 Proof of Theorem 2: Analysis of Finite-Step PSI-Gate Optimization

In this section, we provide the complete statement and proof of Theorem 2. We first derive the closed-form optimal PSI-Gate and then characterize the effect of finite-step optimization from the identity initialization.

Theorem 2 (Identity-preserving bias of finite-step optimization). For task t, consider the PSI surrogate loss

$$\mathcal{L}_{\mathrm{PSI}}^{(t)}(\boldsymbol{G}_t) = \sum_{\substack{i\in\mathcal{T}\\ i\neq t}} \left\|\boldsymbol{G}_t\widehat{\boldsymbol{H}}_i\right\|_F^2 + \left\|\boldsymbol{G}_t\widehat{\boldsymbol{H}}_t - \widehat{\boldsymbol{H}}_t\right\|_F^2. \tag{28}$$

Define $\boldsymbol{C}_i = \widehat{\boldsymbol{H}}_i\widehat{\boldsymbol{H}}_i^\top$ and $\boldsymbol{S} = \sum_{i\in\mathcal{T}}\boldsymbol{C}_i$, and assume that $\boldsymbol{S} \succ \boldsymbol{0}$. Then the PSI surrogate loss is a strictly convex quadratic function of $\boldsymbol{G}_t$ with the unique minimizer

$$\boldsymbol{G}_t^* = \boldsymbol{C}_t\boldsymbol{S}^{-1}. \tag{29}$$

Suppose that $\boldsymbol{G}_t$ is optimized by gradient descent with learning rate ζ from the identity initialization $\boldsymbol{G}_t^{(0)} = \boldsymbol{I}$. After K optimization steps,

$$\boldsymbol{G}_t^{(K)} = \boldsymbol{G}_t^* + (\boldsymbol{I} - \boldsymbol{G}_t^*)(\boldsymbol{I} - 2\zeta\boldsymbol{S})^K. \tag{30}$$

Furthermore, let $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$, where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{d_{\mathrm{in}}})$, and assume $0 < 2\zeta\lambda_{\max}(\boldsymbol{S}) < 1$. Then the residual associated with the jth eigen-direction of $\boldsymbol{S}$ is scaled after K steps by $(1 - 2\zeta\lambda_j)^K$. Hence, the identity-preserving residual decays more slowly along directions with smaller eigenvalues and more rapidly along directions with larger eigenvalues. Moreover,

$$\lim_{K\to\infty} \boldsymbol{G}_t^{(K)} = \boldsymbol{G}_t^*. \tag{31}$$

Proof. We first express the PSI surrogate loss as a quadratic function of the gate and derive its optimal solution. We then analyze the gradient-descent trajectory from the identity initialization and characterize its behavior in the eigenspace of $\boldsymbol{S}$.

1. Quadratic form and optimal PSI-Gate. Using $\|\boldsymbol{A}\|_F^2 = \mathrm{tr}(\boldsymbol{A}\boldsymbol{A}^\top)$, the PSI surrogate loss can be written as

$$\begin{aligned}\mathcal{L}_{\mathrm{PSI}}^{(t)}(\boldsymbol{G}_t) &= \sum_{i\neq t} \mathrm{tr}\left(\boldsymbol{G}_t\boldsymbol{C}_i\boldsymbol{G}_t^\top\right) + \mathrm{tr}\left[(\boldsymbol{G}_t - \boldsymbol{I})\boldsymbol{C}_t(\boldsymbol{G}_t - \boldsymbol{I})^\top\right] \\ &= \mathrm{tr}\left(\boldsymbol{G}_t\boldsymbol{S}\boldsymbol{G}_t^\top\right) - 2\,\mathrm{tr}\left(\boldsymbol{G}_t\boldsymbol{C}_t\right) + \mathrm{tr}(\boldsymbol{C}_t).\end{aligned} \tag{32}$$

Since $\boldsymbol{S} \succ \boldsymbol{0}$, this objective is strictly convex in $\boldsymbol{G}_t$. Its gradient is

$$\nabla_{\boldsymbol{G}_t}\mathcal{L}_{\mathrm{PSI}}^{(t)} = 2(\boldsymbol{G}_t\boldsymbol{S} - \boldsymbol{C}_t). \tag{33}$$

Setting the gradient to zero gives $\boldsymbol{G}_t^*\boldsymbol{S} = \boldsymbol{C}_t$, and therefore

$$\boldsymbol{G}_t^* = \boldsymbol{C}_t\boldsymbol{S}^{-1}. \tag{34}$$

This proves Eq. 29.

2. Finite-step optimization from the identity. Gradient descent with learning rate ζ gives

$$\boldsymbol{G}_t^{(k+1)} = \boldsymbol{G}_t^{(k)} - 2\zeta\left(\boldsymbol{G}_t^{(k)}\boldsymbol{S} - \boldsymbol{C}_t\right). \tag{35}$$

Since $\boldsymbol{C}_t = \boldsymbol{G}_t^* \boldsymbol{S}$, subtracting $\boldsymbol{G}_t^*$ from both sides gives

$$\begin{aligned} \boldsymbol{G}_t^{(k+1)} - \boldsymbol{G}_t^* &= \boldsymbol{G}_t^{(k)} - \boldsymbol{G}_t^* - 2\zeta(\boldsymbol{G}_t^{(k)} - \boldsymbol{G}_t^*)\boldsymbol{S} \\ &= (\boldsymbol{G}_t^{(k)} - \boldsymbol{G}_t^*)(\boldsymbol{I} - 2\zeta \boldsymbol{S}). \end{aligned} \tag{36}$$

Applying this recurrence for K steps yields

$$\boldsymbol{G}_t^{(K)} - \boldsymbol{G}_t^* = (\boldsymbol{G}_t^{(0)} - \boldsymbol{G}_t^*)(\boldsymbol{I} - 2\zeta \boldsymbol{S})^K. \tag{37}$$

Using $\boldsymbol{G}_t^{(0)} = \boldsymbol{I}$, we obtain

$$\boldsymbol{G}_t^{(K)} = \boldsymbol{G}_t^* + (\boldsymbol{I} - \boldsymbol{G}_t^*)(\boldsymbol{I} - 2\zeta \boldsymbol{S})^K, \tag{38}$$

which proves Eq. 30.

3. Spectral behavior of the identity-preserving residual. Let $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$, where $\boldsymbol{Q}$ is orthogonal. Then

$$(\boldsymbol{I} - 2\zeta \boldsymbol{S})^K = \boldsymbol{Q}(\boldsymbol{I} - 2\zeta \boldsymbol{\Lambda})^K \boldsymbol{Q}^\top. \tag{39}$$

For an eigenvector $\boldsymbol{q}_j$ of $\boldsymbol{S}$ with eigenvalue λ_j,

$$(\boldsymbol{I} - 2\zeta \boldsymbol{S})^K \boldsymbol{q}_j = (1 - 2\zeta\lambda_j)^K \boldsymbol{q}_j. \tag{40}$$

Thus, the component of the residual $\boldsymbol{G}_t^{(K)} - \boldsymbol{G}_t^*$ associated with this eigen-direction is scaled by $(1 - 2\zeta\lambda_j)^K$.

Under $0 < 2\zeta\lambda_{\max}(\boldsymbol{S}) < 1$, we have $0 < 1 - 2\zeta\lambda_j < 1$ for every j. For a fixed K, this factor is larger for smaller λ_j and smaller for larger λ_j. The identity-preserving residual therefore decays more slowly along low-eigenvalue directions and more rapidly along high-eigenvalue directions.

4. Convergence to the optimal PSI-Gate. Under the same step-size condition,

$$\lim_{K\to\infty} (\boldsymbol{I} - 2\zeta \boldsymbol{S})^K = \boldsymbol{0}. \tag{41}$$

Substituting this result into Eq. 30 gives

$$\lim_{K\to\infty} \boldsymbol{G}_t^{(K)} = \boldsymbol{G}_t^*, \tag{42}$$

which proves Eq. 31 and completes the proof. □

B ALGORITHM

Algorithm 1 Matrix-Wise Gated Task Arithmetic (GTA)

Input: Base weights $\boldsymbol{W}_{\text{base}}$, task-specific weights $\{\boldsymbol{W}_t\}_{t\in\mathcal{T}}$, number of gate updates K, gate learning rate ζ
Output: Merged weights $\boldsymbol{W}_m$

for all $t \in \mathcal{T}$ **do**
  $\boldsymbol{\tau}_t \leftarrow \boldsymbol{W}_t - \boldsymbol{W}_{\text{base}}$
  Compute the compact SVD $\boldsymbol{\tau}_t = \boldsymbol{U}_t \boldsymbol{\Sigma}_t \boldsymbol{V}_t^\top$
  $\widehat{\boldsymbol{H}}_t \leftarrow \boldsymbol{V}_t \boldsymbol{\Sigma}_t$
  $\boldsymbol{G}_t^{(0)} \leftarrow \boldsymbol{I}_{d_{\text{in}}}$
end for
for all $t \in \mathcal{T}$ **do**
  for $k = 0, \ldots, K-1$ **do**
    $\mathcal{L}_{\text{PSI}}^{(t)} \leftarrow \sum_{\substack{i\in\mathcal{T}\\ i\neq t}} \left\| \boldsymbol{G}_t^{(k)} \widehat{\boldsymbol{H}}_i \right\|_F^2 + \left\| \boldsymbol{G}_t^{(k)} \widehat{\boldsymbol{H}}_t - \widehat{\boldsymbol{H}}_t \right\|_F^2$
    Update $\boldsymbol{G}_t^{(k)}$ to $\boldsymbol{G}_t^{(k+1)}$ using Adam with learning rate ζ to minimize $\mathcal{L}_{\text{PSI}}^{(t)}$
  end for
end for
$\boldsymbol{\tau}_m^{(\text{GTA})} \leftarrow \sum_{t\in\mathcal{T}} \boldsymbol{\tau}_t \boldsymbol{G}_t^{(K)}$
$\boldsymbol{W}_m \leftarrow \boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_m^{(\text{GTA})}$
return $\boldsymbol{W}_m$

Algorithm 1 summarizes GTA for a single weight matrix; the same procedure is applied independently to each matrix being merged. We first compute the task vectors and their surrogate input representations $\widehat{\boldsymbol{H}}_t = \boldsymbol{V}_t \boldsymbol{\Sigma}_t$, and initialize each PSI-Gate with the identity matrix. Keeping the task vectors and surrogate input representations fixed, we optimize each PSI-Gate for K steps using Adam and the PSI surrogate loss in Eq. 6. Finally, each learned PSI-Gate is absorbed into its corresponding task vector by right multiplication, and the resulting gated task vectors are summed and added to the base weights. The entire procedure uses only model weights and requires no additional data.

C MODELS & DATASETS

We first examine merging performance as the number of vision tasks increases and then extend the evaluation to natural language classification and reasoning tasks that require generating reasoning steps. Together, these settings cover diverse task types, model architectures, and model sizes.

Model Preparation For vision, we use task-specific checkpoints provided by *FusionBench* (Tang et al., 2025) without retraining: ViT-B/32, ViT-B/16, and ViT-L/14 (Dosovitskiy et al., 2021). For natural language classification and reasoning, we construct LLaMA (Grattafiori et al., 2024) and Qwen2 (Yang et al., 2024a) checkpoints, respectively, through full-parameter supervised fine-tuning (SFT) with AdamW, without parameter-efficient methods such as LoRA.

Relation to the SGD Assumption Theorem 1 provides an exact characterization under plain SGD, whereas our LLaMA and Qwen2 checkpoints are fine-tuned using AdamW. We therefore use the theorem as a theoretical characterization under an idealized optimization setting. Deviations from the SGD assumption are characterized by the residual formulation in Appendix A.1.1.

Vision Tasks: ViT We evaluate vision model merging with 8, 14, and 20 tasks. The task sets are nested: each larger set contains all datasets in the preceding set and adds new ones, allowing us to examine merging performance as the number of tasks increases. We use the same task sets for all three ViT models. The datasets are listed below.

8 Tasks: SUN397 (Xiao et al., 2010), Stanford Cars (Krause et al., 2013), RESISC45 (Cheng et al., 2017), EuroSAT (Helber et al., 2018), SVHN (Netzer et al., 2011), GTSRB (Stallkamp et al., 2012), MNIST (Lecun et al., 1998), DTD (Cimpoi et al., 2014).

14 Tasks: 8 Tasks + Oxford Flowers102 (Nilsback & Zisserman, 2008), PCAM (Veeling et al., 2018), FER2013 (Goodfellow et al., 2013), Oxford-IIIT Pet (Parkhi et al., 2012), STL10 (Coates et al., 2011), CIFAR100 (Krizhevsky, 2009).

20 Tasks: 14 Tasks + CIFAR10 (Krizhevsky, 2009), Food101 (Bossard et al., 2014), FashionMNIST (Xiao et al., 2017), EMNIST Letters (Cohen et al., 2017), KMNIST (Clanuwat et al., 2018), Rendered SST2 (Menon & Vondrick, 2022).

Natural Language Classification Tasks: LLaMA To examine merging performance across different model sizes, we use LLaMA-3.2-1B, LLaMA-3.2-3B, and LLaMA-3.1-8B. For each base model, we perform full-parameter SFT with AdamW separately on the same 8 natural language classification tasks and merge only the task-specific models derived from the same base model. The datasets are MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), AG News (Zhang et al., 2016), CommonsenseQA (Talmor et al., 2019), MNLI (Williams et al., 2018), WinoGrande (Sakaguchi et al., 2019), MRPC (Dolan & Brockett, 2005), and WiC (Pilehvar & Camacho-Collados, 2019). We evaluate task performance by predicting the correct label or answer choice.

Natural Language Reasoning Tasks: Qwen2 To extend the evaluation beyond classification to tasks that require generating reasoning steps, we use 5 datasets containing chain-of-thought (CoT) reasoning: GSM8K (Cobbe et al., 2021), StrategyQA (Geva et al., 2021), OpenBookQA (Mihaylov et al., 2018), e-SNLI (Camburu et al., 2018), and Explanations for CommonsenseQA (Aggarwal et al., 2021). We perform full-parameter SFT of Qwen2-7B with AdamW separately on each dataset and merge the resulting 5 task-specific models. Unlike the classification settings, we extract the final answer from the generated text and evaluate it using exact match against the reference answer. This setting evaluates whether merging methods preserve the performance gains obtained by fine-tuning on generation tasks involving reasoning.

D MERGING RESULTS

Method	MMLU	HS	AGN	CQA	MNLI	WG	MRPC	WiC	Avg.
Non-Merged									
Base	27.65	25.03	31.32	21.29	35.45	49.42	66.49	50.00	38.33
Fine-Tuned	49.28	84.45	95.03	71.74	88.58	70.48	86.72	71.79	77.26
Merged									
TA	38.06 (48.1%)	38.02 (21.9%)	77.38 (72.3%)	51.52 (59.9%)	37.05 (3.0%)	55.42 (28.5%)	66.96 (2.3%)	50.21 (1.0%)	51.83 (34.7%)
TIES	39.32 (54.0%)	50.09 (42.2%)	76.13 (70.3%)	52.50 (61.9%)	43.12 (14.4%)	60.91 (54.5%)	68.75 (11.2%)	52.79 (12.8%)	55.45 (44.0%)
TSV	43.53 (73.4%)	70.17 (76.0%)	87.34 (87.9%)	66.34 (89.3%)	48.27 (24.1%)	65.21 (75.0%)	70.43 (19.5%)	53.86 (17.7%)	63.14 (63.7%)
Subspace Boosting	44.06 (75.9%)	67.47 (71.4%)	87.21 (87.7%)	65.36 (87.3%)	46.81 (21.4%)	63.33 (66.0%)	69.74 (16.0%)	53.21 (14.8%)	62.15 (61.2%)
WUDI	42.88 (70.4%)	74.38 (83.1%)	88.32 (89.5%)	66.50 (89.6%)	53.88 (34.7%)	65.51 (76.4%)	75.59 (45.0%)	59.36 (43.0%)	65.80 (70.6%)
ACTMat	44.37 (77.3%)	74.47 (83.2%)	90.38 (92.7%)	64.62 (85.9%)	53.95 (34.8%)	68.11 (88.8%)	73.22 (33.2%)	56.00 (27.5%)	65.64 (70.2%)
GTA	**43.96 (75.4%)**	**74.73 (83.6%)**	**90.79 (93.3%)**	**66.42 (89.4%)**	**59.62 (45.5%)**	**68.92 (92.6%)**	**75.01 (42.1%)**	**57.29 (33.4%)**	**67.09 (73.9%)**

Table 4: Model merging performance on eight natural language classification tasks using LLaMA-3.2-1B. Each task column reports accuracy (%), while *Avg.* reports the average accuracy over all eight tasks. Gray values in parentheses indicate the Task Recovery Rate (TRR, %). For *Avg.*, TRR is computed using the average accuracies of the merged, base, and task-specific fine-tuned models. *Base* and *Fine-Tuned* are shown as non-merged references.

Method	MMLU	HS	AGN	CQA	MNLI	WG	MRPC	WiC	Avg.
Non-Merged									
Base	57.13	49.03	64.16	63.47	38.13	49.61	66.49	50.07	54.76
Fine-Tuned	69.02	96.05	95.53	82.31	92.08	88.62	88.70	75.64	85.99
Merged									
TA	60.87 (31.4%)	56.22 (15.3%)	79.12 (47.7%)	71.09 (40.4%)	45.15 (13.0%)	56.26 (17.1%)	67.77 (5.7%)	60.57 (41.1%)	62.13 (23.6%)
TIES	48.09 (-76.0%)	52.30 (6.9%)	88.46 (77.5%)	63.88 (2.2%)	42.80 (8.7%)	80.90 (80.2%)	74.09 (34.2%)	49.14 -(3.6%)	62.46 (24.6%)
TSV	59.78 (22.3%)	88.90 (84.8%)	93.55 (93.7%)	79.20 (83.5%)	81.70 (80.8%)	86.64 (94.9%)	81.68 (68.4%)	51.00 (3.6%)	77.81 (73.8%)
Subspace Boosting	59.89 (23.2%)	90.54 (88.3%)	94.18 (95.7%)	79.36 (84.3%)	80.19 (78.0%)	87.08 (96.1%)	80.17 (61.6%)	51.79 (6.7%)	77.90 (74.1%)
WUDI	56.33 (-6.7%)	91.45 (90.2%)	93.66 (94.0%)	79.85 (87.0%)	75.00 (68.3%)	87.42 (96.9%)	85.45 (85.4%)	62.36 (48.0%)	78.94 (77.4%)
ACTMat	61.41 (36.0%)	92.45 (92.4%)	94.64 (97.2%)	81.41 (95.2%)	86.46 (89.6%)	87.73 (97.7%)	85.16 (84.1%)	61.86 (46.1%)	81.39 (85.3%)
GTA	**61.57 (37.4%)**	**92.91 (93.3%)**	**95.07 (98.5%)**	**82.56 (101.3%)**	**89.21 (94.7%)**	**88.82 (100.5%)**	**85.22 (84.3%)**	**58.86 (34.4%)**	**81.78 (86.5%)**

Table 5: Model merging performance on eight natural language classification tasks using LLaMA-3.1-8B. Each task column reports accuracy (%), while *Avg.* reports the average accuracy over all eight tasks. Gray values in parentheses indicate the Task Recovery Rate (TRR, %). For *Avg.*, TRR is computed using the average accuracies of the merged, base, and task-specific fine-tuned models. *Base* and *Fine-Tuned* are shown as non-merged references.

Tables 4, 5, and 6 provide the task-level results for all three evaluated LLaMA model sizes. Relative to ACTMat, GTA improves average accuracy by 1.45 percentage points on LLaMA-3.2-1B, is effectively tied on LLaMA-3.2-3B (78.07 vs. 78.09), and improves average accuracy by 0.39 percentage points on LLaMA-3.1-8B. The corresponding average TRRs are 73.9%, 85.5%, and 86.5% for GTA, compared with 70.2%, 85.5%, and 85.3% for ACTMat. This comparison is particularly informative

Method	MMLU	HS	AGN	CQA	MNLI	WG	MRPC	WiC	Avg.
Non-Merged									
Base	30.77	27.08	63.55	51.35	35.42	51.17	64.41	49.79	46.69
Fine-Tuned	61.32	93.03	95.16	79.28	90.37	85.82	87.01	75.21	83.40
Merged									
TA	53.08 (73.0%)	63.90 (55.8%)	85.61 (69.8%)	68.47 (61.3%)	47.92 (22.7%)	59.87 (25.1%)	66.84 (10.8%)	51.21 (5.6%)	62.11 (42.0%)
TIES	51.87 (69.1%)	70.70 (66.1%)	85.43 (69.2%)	71.33 (71.6%)	40.54 (9.3%)	73.80 (65.3%)	67.88 (15.4%)	47.93 (-7.3%)	63.69 (46.3%)
TSV	54.85 (78.8%)	83.05 (84.9%)	91.17 (87.4%)	75.43 (86.2%)	72.66 (67.8%)	80.68 (85.2%)	75.83 (50.5%)	53.57 (14.9%)	73.40 (72.8%)
Subspace Boosting	55.88 (82.2%)	84.41 (86.9%)	91.62 (88.8%)	75.84 (87.7%)	74.56 (71.2%)	79.90 (82.9%)	71.83 (32.8%)	53.50 (14.6%)	73.44 (72.9%)
WUDI	54.47 (77.6%)	87.48 (91.6%)	91.30 (87.8%)	76.49 (90.0%)	77.86 (77.2%)	82.39 (90.1%)	82.32 (79.2%)	62.36 (49.4%)	76.83 (82.1%)
ACTMat	56.30 (83.6%)	88.53 (93.2%)	92.58 (91.8%)	76.00 (88.3%)	81.54 (83.9%)	82.08 (89.2%)	82.61 (80.5%)	65.07 (60.1%)	78.09 (85.5%)
GTA	**56.24 (83.4%)**	**89.13 (94.1%)**	**93.04 (93.3%)**	**77.48 (93.5%)**	**79.40 (80.0%)**	**84.36 (95.8%)**	**82.96 (82.1%)**	**61.93 (47.8%)**	**78.07 (85.5%)**

Table 6: Model merging performance on eight natural language classification tasks using LLaMA-3.2-3B. Each task column reports accuracy (%), while *Avg.* reports the average accuracy over all eight tasks. Gray values in parentheses indicate the Task Recovery Rate (TRR, %). For *Avg.*, TRR is computed using the average accuracies of the merged, base, and task-specific fine-tuned models. *Base* and *Fine-Tuned* are shown as non-merged references.

because, when task-vector Gram matrices are used as covariance estimates, the closed-form ACT-Mat merge coincides with the limiting GTA solution obtained as $K \to \infty$ in Theorem 2. Across the evaluated ViT, LLaMA, and Qwen2 models, finite-step GTA is consistently competitive with, and often outperforms, this limiting solution. These results are consistent with the finite-step perspective of Theorem 2: optimization retains an identity-preserving residual rather than immediately reaching the limiting solution, but the theorem does not predict downstream accuracy.

E Interference Analysis Results

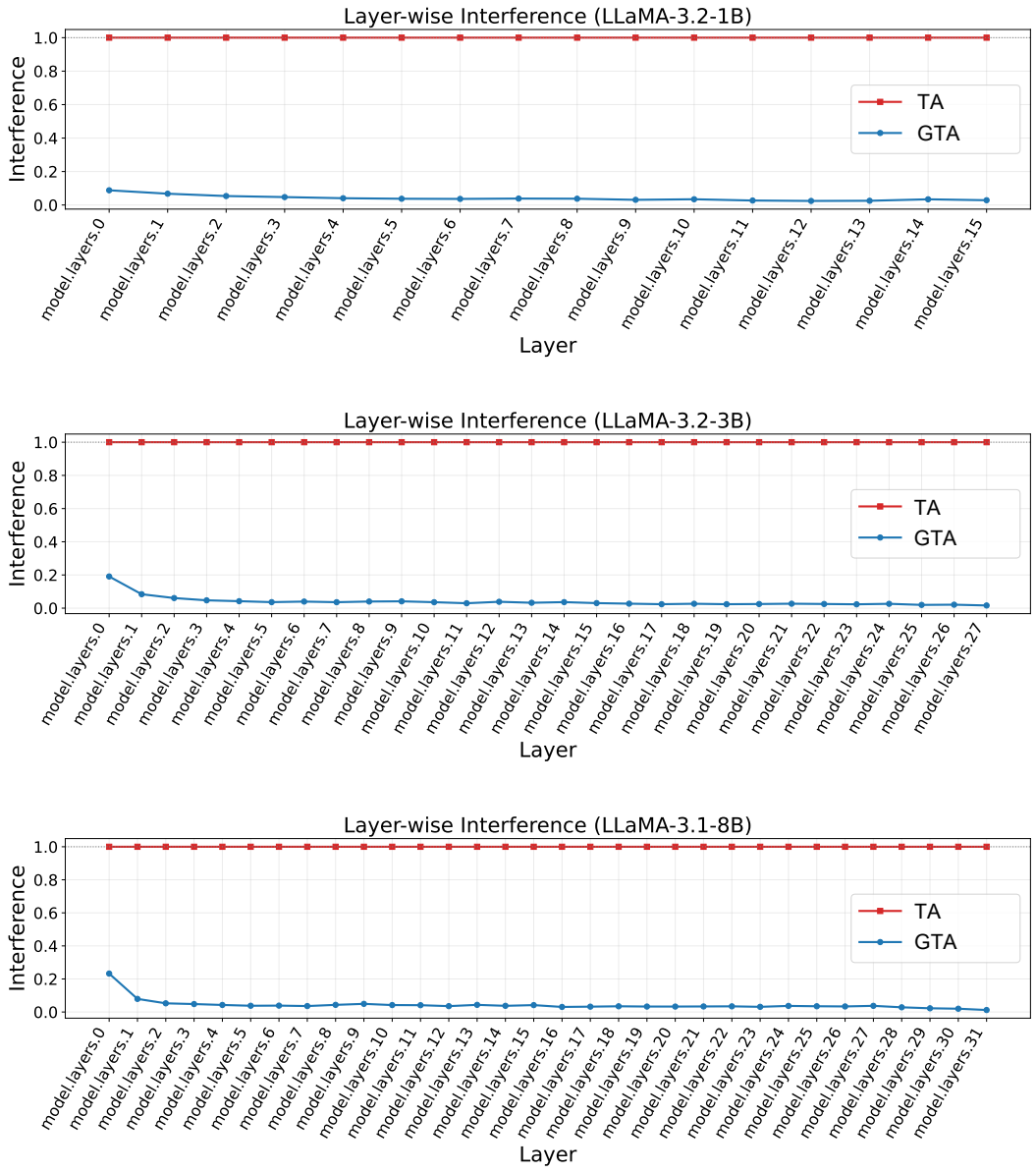


Figure 4: Layer-level task interference on real input activations for LLaMA-3.2-1B (top), LLaMA-3.2-3B (middle), and LLaMA-3.1-8B (bottom) when merging eight natural language classification tasks. Measurements use 100 samples per task with a fixed random seed of 42. For each model, interference is normalized by the ungated Task Arithmetic reference: "TA" corresponds to $\mathcal{I}^{(\mathrm{TA})}/\mathcal{I}^{(\mathrm{TA})} = 1$, while "GTA" corresponds to $\mathcal{I}^{(\mathrm{GTA})}/\mathcal{I}^{(\mathrm{TA})}$. Values are averaged within each layer for visualization, and values below 1 indicate reduced interference after gating.

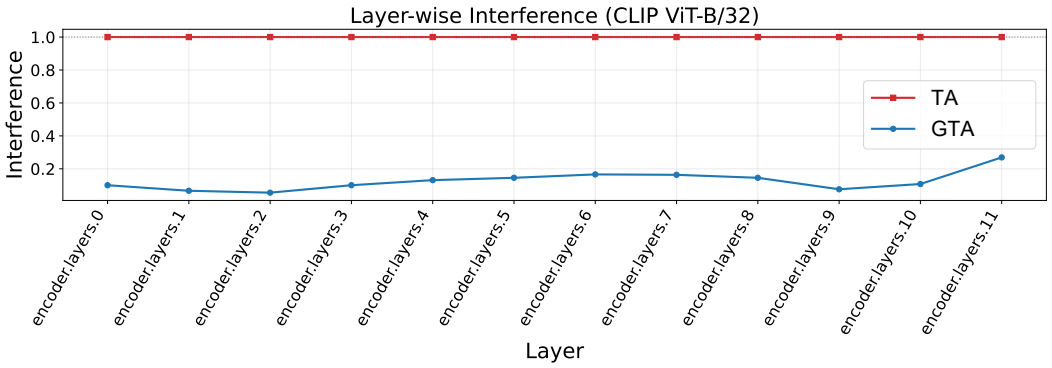


Figure 5: Layer-level task interference on real input activations for CLIP ViT-B/32 when merging eight vision classification tasks. CLIP ViT-B/32 is evaluated as a representative setting for the CLIP model merging experiments. Measurements use 100 samples per task with a fixed random seed of 42. Interference is normalized by the ungated Task Arithmetic reference: "TA" corresponds to $\mathcal{I}^{(\mathrm{TA})}/\mathcal{I}^{(\mathrm{TA})} = 1$, while "GTA" corresponds to $\mathcal{I}^{(\mathrm{GTA})}/\mathcal{I}^{(\mathrm{TA})}$. Values are averaged within each layer for visualization, and values below 1 indicate reduced interference after gating.

Figures 4 and 5 compare layer-level task interference before and after gating on real input activations for LLaMA-3.2-1B, LLaMA-3.2-3B, LLaMA-3.1-8B, and CLIP ViT-B/32. CLIP ViT-B/32 serves as a representative vision model, while the three LLaMA models allow us to examine the effect across different language-model sizes.

For each model, we normalize the ungated Task Arithmetic interference $\mathcal{I}^{(\mathrm{TA})}$ to 1 and report the GTA interference as $\mathcal{I}^{(\mathrm{GTA})}/\mathcal{I}^{(\mathrm{TA})}$. Thus, a value below 1 indicates reduced layer-level task interference after applying the PSI-Gates.

Across all evaluated models, GTA consistently reduces the normalized interference relative to the ungated Task Arithmetic reference. The consistent reduction across vision and language models, different model sizes, and different task types shows that PSI-Gates learned from $\boldsymbol{V}_t\boldsymbol{\Sigma}_t$ can suppress cross-task interference on real input activations, despite being learned solely from task-vector-derived surrogate representations.

F PRESERVATION OF INDIVIDUAL TASK-VECTOR PERFORMANCE

Method	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Original									
$\boldsymbol{W}_{\text{base}}$	63.18	59.58	60.27	45.00	31.63	32.53	48.26	44.20	48.08
$\boldsymbol{\tau}_t$	74.86	78.52	95.14	99.07	97.27	98.91	99.58	79.68	90.38
Gated									
$\boldsymbol{\tau}_t\boldsymbol{G}_t$	75.31	76.98	94.00	98.15	95.35	97.64	99.42	76.49	89.17
	(103.8%)	(91.9%)	(96.7%)	(98.3%)	(97.1%)	(98.1%)	(99.7%)	(91.0%)	(97.1%)
	+0.45	-1.54	-1.14	-0.93	-1.92	-1.27	-0.16	-3.19	-1.21

Table 7: Individual task-vector performance before and after applying the learned PSI-Gates on CLIP ViT-B/32, using the eight-task setting as a representative case for the vision merging experiments. For each task t, $\boldsymbol{\tau}_t$ denotes the original task vector, while $\boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}$ denotes its gated counterpart. Each entry reports accuracy (%) on task t when the corresponding task vector is applied individually to the base model. Gray values in parentheses indicate the Task Recovery Rate (TRR, %), computed using $\boldsymbol{W}_{\text{base}}$, $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t$, and $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}$, with the original task vector serving as the task-specific reference. The bottom row reports the accuracy difference $\text{ACC}_t(\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}) - \text{ACC}_t(\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t)$.

Method	MMLU	HS	AGN	CQA	MNLI	WG	MRPC	WiC	Avg.
Original									
$\boldsymbol{W}_{\text{base}}$	27.65	25.03	31.32	21.29	35.45	49.42	66.49	50.00	38.33
$\boldsymbol{\tau}_t$	49.28	84.45	95.03	71.74	88.58	70.48	86.72	71.79	77.26
Gated									
$\boldsymbol{\tau}_t\boldsymbol{G}_t$	46.55	80.98	95.22	67.49	84.36	70.68	82.43	67.93	74.46
	(87.4%)	(94.2%)	(100.3%)	(91.6%)	(92.1%)	(100.9%)	(78.8%)	(82.3%)	(92.8%)
	-2.73	-3.47	+0.20	-4.26	-4.22	+0.20	-4.29	-3.86	-2.80

Table 8: Individual task-vector performance before and after applying the learned PSI-Gates on LLaMA-3.2-1B. For each task t, $\boldsymbol{\tau}_t$ denotes the original task vector, while $\boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}$ denotes its gated counterpart. Each entry reports accuracy (%) on task t when the corresponding task vector is applied individually to the base model. Gray values in parentheses indicate the Task Recovery Rate (TRR, %), computed using $\boldsymbol{W}_{\text{base}}$, $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t$, and $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}$, with the original task vector serving as the task-specific reference. The bottom row reports the accuracy difference $\text{ACC}_t(\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}) - \text{ACC}_t(\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t)$.

Method	MMLU	HS	AGN	CQA	MNLI	WG	MRPC	WiC	Avg.
Original									
$\boldsymbol{W}_{\text{base}}$	30.77	27.08	63.55	51.35	35.42	51.17	64.41	49.79	46.69
$\boldsymbol{\tau}_t$	61.32	93.03	95.16	79.28	90.37	85.82	87.01	75.21	83.40
Gated									
$\boldsymbol{\tau}_t\boldsymbol{G}_t$	59.63	92.10	95.18	77.31	87.23	84.18	86.49	75.29	82.18
	(94.5%)	(98.6%)	(100.1%)	(93.0%)	(94.3%)	(95.3%)	(97.7%)	(100.3%)	(96.7%)
	-1.69	-0.93	+0.03	-1.97	-3.14	-1.64	-0.52	+0.07	-1.22

Table 9: Individual task-vector performance before and after applying the learned PSI-Gates on LLaMA-3.2-3B. For each task t, $\boldsymbol{\tau}_t$ denotes the original task vector, while $\boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}$ denotes its gated counterpart. Each entry reports accuracy (%) on task t when the corresponding task vector is applied individually to the base model. Gray values in parentheses indicate the Task Recovery Rate (TRR, %), computed using $\boldsymbol{W}_{\text{base}}$, $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t$, and $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}$, with the original task vector serving as the task-specific reference. The bottom row reports the accuracy difference $\text{ACC}_t(\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}) - \text{ACC}_t(\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t)$.

Method	MMLU	HS	AGN	CQA	MNLI	WG	MRPC	WiC	Avg.
Original									
$\boldsymbol{W}_{\text{base}}$	57.13	49.03	64.16	63.47	38.13	49.61	66.49	50.07	54.76
$\boldsymbol{\tau}_t$	69.02	96.05	95.53	82.31	92.08	88.62	88.70	75.64	85.99
Gated									
$\boldsymbol{\tau}_t\boldsymbol{G}_t$	68.70	95.55	95.55	83.46	91.49	88.66	88.29	73.57	85.66
	(97.3%)	(98.9%)	(100.1%)	(106.1%)	(98.9%)	(100.1%)	(98.2%)	(91.9%)	(98.9%)
	-0.32	-0.50	+0.03	+1.15	-0.59	+0.04	-0.41	-2.07	-0.33

Table 10: Individual task-vector performance before and after applying the learned PSI-Gates on LLaMA-3.1-8B. For each task t, $\boldsymbol{\tau}_t$ denotes the original task vector, while $\boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}$ denotes its gated counterpart. Each entry reports accuracy (%) on task t when the corresponding task vector is applied individually to the base model. Gray values in parentheses indicate the Task Recovery Rate (TRR, %), computed using $\boldsymbol{W}_{\text{base}}$, $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t$, and $\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}$, with the original task vector serving as the task-specific reference. The bottom row reports the accuracy difference $\text{ACC}_t(\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t\boldsymbol{G}_t^{(K)}) - \text{ACC}_t(\boldsymbol{W}_{\text{base}} + \boldsymbol{\tau}_t)$.

Tables 7–10 extend the analysis of individual task-vector preservation to the remaining evaluated models. Across CLIP ViT-B/32 and LLaMA models of different sizes, the performance of individual task vectors generally remains close to its original level after applying the PSI-Gates. The effect varies across tasks: some show small decreases, while others perform comparably to or slightly better than their original task vectors. Overall, these results show that the PSI-Gates largely preserve the task-specific capabilities acquired through fine-tuning.

G Finite-Step Gate Optimization and Optimization-Budget Robustness

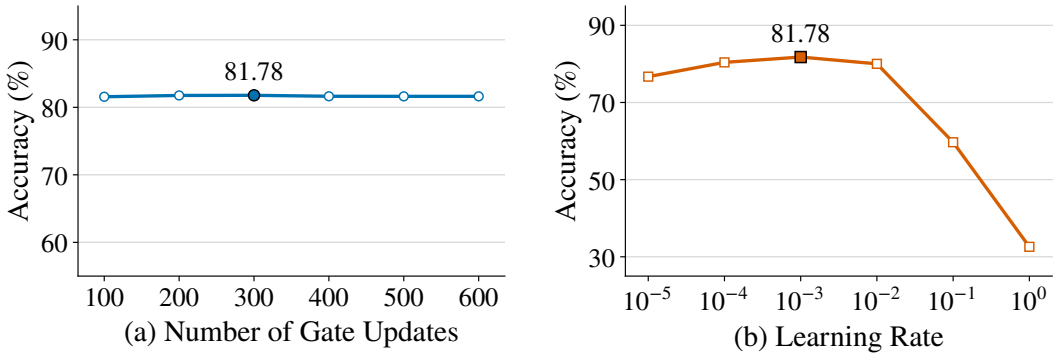


Figure 6: Downstream behavior of finite-step PSI-Gate optimization on LLaMA-3.1-8B. The left panel varies the number of gate updates K with the learning rate fixed at $\zeta = 10^{-3}$, while the right panel varies ζ with $K = 300$. Filled markers indicate the default finite-step configuration used in our experiments.

Theorem 2 shows that the number of gate updates K and the learning rate ζ control the identity-preserving residual that remains after finite-step optimization. Thus, these quantities are not only numerical hyperparameters: together, they determine how far a learned PSI-Gate moves from its identity initialization toward the optimum of the PSI surrogate loss. Figure 6 examines the downstream behavior of this finite-step optimization regime on LLaMA-3.1-8B.

With $\zeta = 10^{-3}$, varying K from 100 to 600 keeps the average accuracy within a narrow range of approximately 81–82%. The default setting of $K = 300$ achieves 81.78%, while increasing the number of updates provides no systematic improvement over the evaluated range. These results indicate that strong merging performance does not require pushing the gate optimization as far as possible. Instead, the default budget lies in a broad and stable finite-step regime.

With $K = 300$, accuracy reaches its highest observed value of 81.78% at $\zeta = 10^{-3}$ and remains relatively stable from 10^{-5} to 10^{-2}. In contrast, performance decreases substantially at 10^{-1} and further at 10^{0}, showing that overly aggressive gate updates can be detrimental. Together, these results support $K = 300$ and $\zeta = 10^{-3}$ as a stable finite-step configuration that avoids unnecessarily aggressive optimization.

Our experiments use Adam, whereas Theorem 2 analyzes gradient descent. Therefore, the observed behavior should be viewed as qualitative empirical support for the finite-step perspective rather than as a direct verification of the theorem's exact spectral dynamics. Appendix H further examines how the retained influence of identity initialization affects the suppression–preservation trade-off.

H GATE INITIALIZATION AND THE SUPPRESSION–PRESERVATION TRADE-OFF

Theorem 2 shows that finite-step gradient descent from $\boldsymbol{G}_t^{(0)} = \boldsymbol{I}$ retains an identity-preserving residual whose decay depends on the spectrum of $\boldsymbol{S}$. We empirically examine the consequences of this initialization-dependent bias on LLaMA-3.1-8B and LLaMA-3.2-1B by comparing Identity with two alternatives while keeping all other GTA settings fixed. Constant assigns the same value to every gate entry, whereas Random samples each entry using the entry-wise statistics of the surrogate input representation $\widehat{\boldsymbol{H}}_t = \boldsymbol{V}_t \boldsymbol{\Sigma}_t$. The three initialization schemes are defined as follows.

$$\begin{aligned} \text{Identity:} \quad & \boldsymbol{G}_t^{(0)} = \boldsymbol{I}, \\ \text{Constant:} \quad & \boldsymbol{G}_t^{(0)} = \mathrm{Mean}\Big(|\widehat{\boldsymbol{H}}_t|\Big)\, \mathbf{1}_{d_{\mathrm{in}} \times d_{\mathrm{in}}}, \\ \text{Random:} \quad & [\boldsymbol{G}_t^{(0)}]_{ij} = \mathrm{Mean}(\widehat{\boldsymbol{H}}_t) + \mathrm{Std}(\widehat{\boldsymbol{H}}_t) R_{ij}, \qquad R_{ij} \sim \mathcal{N}(0,1). \end{aligned} \tag{43}$$

Here, $\mathbf{1}_{d_{\mathrm{in}} \times d_{\mathrm{in}}}$ denotes the all-ones matrix, and $\mathrm{Mean}(\cdot)$ and $\mathrm{Std}(\cdot)$ denote the mean and standard deviation over all entries of their matrix argument. Identity applies no initial filtering, so $\boldsymbol{\tau}_t \boldsymbol{G}_t^{(0)} = \boldsymbol{\tau}_t$. In contrast, Constant and Random construct the initial PSI-Gate from the magnitude or distributional statistics of $\widehat{\boldsymbol{H}}_t$, respectively.

Under the assumptions of Theorem 2, let $\boldsymbol{G}_t^*$ denote the optimal PSI-Gate and define $\boldsymbol{S} = \sum_{i \in \mathcal{T}} \widehat{\boldsymbol{H}}_i \widehat{\boldsymbol{H}}_i^\top$. The finite-step expression extends to an arbitrary initialization as

$$\boldsymbol{G}_t^{(K)} = \boldsymbol{G}_t^* + (\boldsymbol{G}_t^{(0)} - \boldsymbol{G}_t^*)(\boldsymbol{I} - 2\zeta \boldsymbol{S})^K. \tag{44}$$

Thus, the three schemes differ through their initial residual $\boldsymbol{G}_t^{(0)} - \boldsymbol{G}_t^*$, while the spectrum of $\boldsymbol{S}$ determines how quickly that residual decays. The Identity case recovers Eq. 7; in particular, initialization-dependent effects persist longer along directions associated with smaller eigenvalues of $\boldsymbol{S}$. Although the experiments below use Adam, the analysis in Theorem 2 provides a useful theoretical perspective on how finite-step optimization can retain initialization-dependent effects.

To distinguish how initialization affects the two roles of a PSI-Gate, we track the task-preservation and interference-suppression components of the PSI surrogate loss separately:

$$\begin{aligned} \mathcal{L}_{\mathrm{pre}}^{(t)} &= \Big\| \boldsymbol{G}_t \widehat{\boldsymbol{H}}_t - \widehat{\boldsymbol{H}}_t \Big\|_F^2, \\ \mathcal{L}_{\mathrm{sup}}^{(t)} &= \sum_{i \in \mathcal{T},\, i \neq t} \Big\| \boldsymbol{G}_t \widehat{\boldsymbol{H}}_i \Big\|_F^2. \end{aligned} \tag{45}$$

The task-preservation loss measures the distortion of task t's surrogate input representation, whereas the interference-suppression loss measures the residual energy of the other tasks' surrogate representations after gating. Lower values indicate closer satisfaction of the corresponding ideal PSI-Gate conditions in Eq. 3. Under Identity initialization, $\boldsymbol{G}_t^{(0)} \widehat{\boldsymbol{H}}_t = \widehat{\boldsymbol{H}}_t$, and hence $\mathcal{L}_{\mathrm{pre}}^{(t)} = 0$ at the beginning of optimization. Neither Constant nor Random has this exact initial preservation property.

We track these loss components for both model sizes. Because LLaMA-3.2-1B exhibits the same qualitative trajectories as LLaMA-3.1-8B, we present the loss curves only for LLaMA-3.1-8B to avoid redundant figures. Downstream accuracies for both models are reported below.

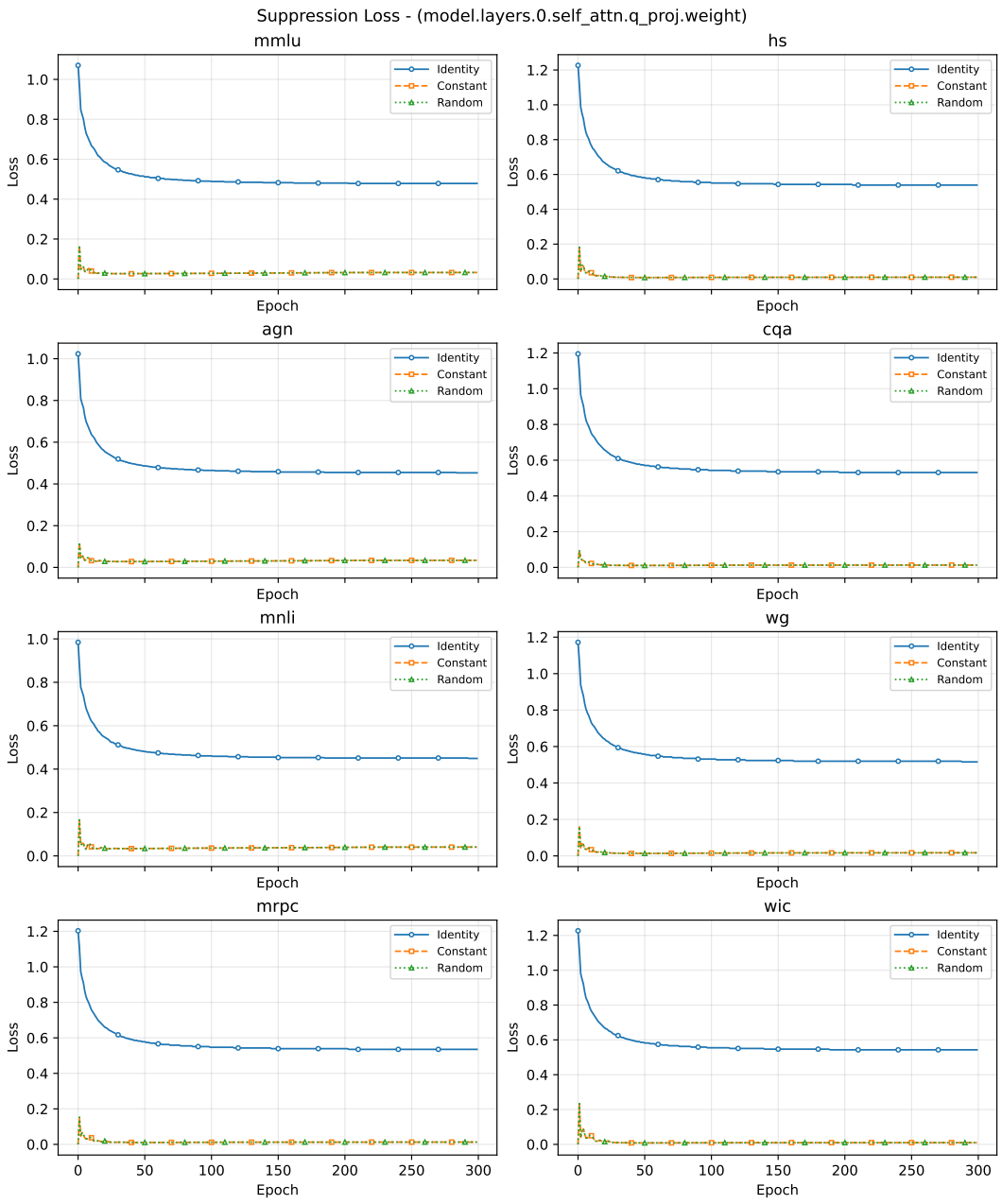


Figure 7: Suppression-loss trajectories for different gate initialization schemes on LLaMA-3.1-8B. Each panel corresponds to one task and reports the loss over 300 gate-update steps for `model.layers.0.self_attn.q_proj.weight`. Identity initialization generally yields a higher suppression loss than Constant and Random, indicating weaker suppression of surrogate input representations from other tasks.

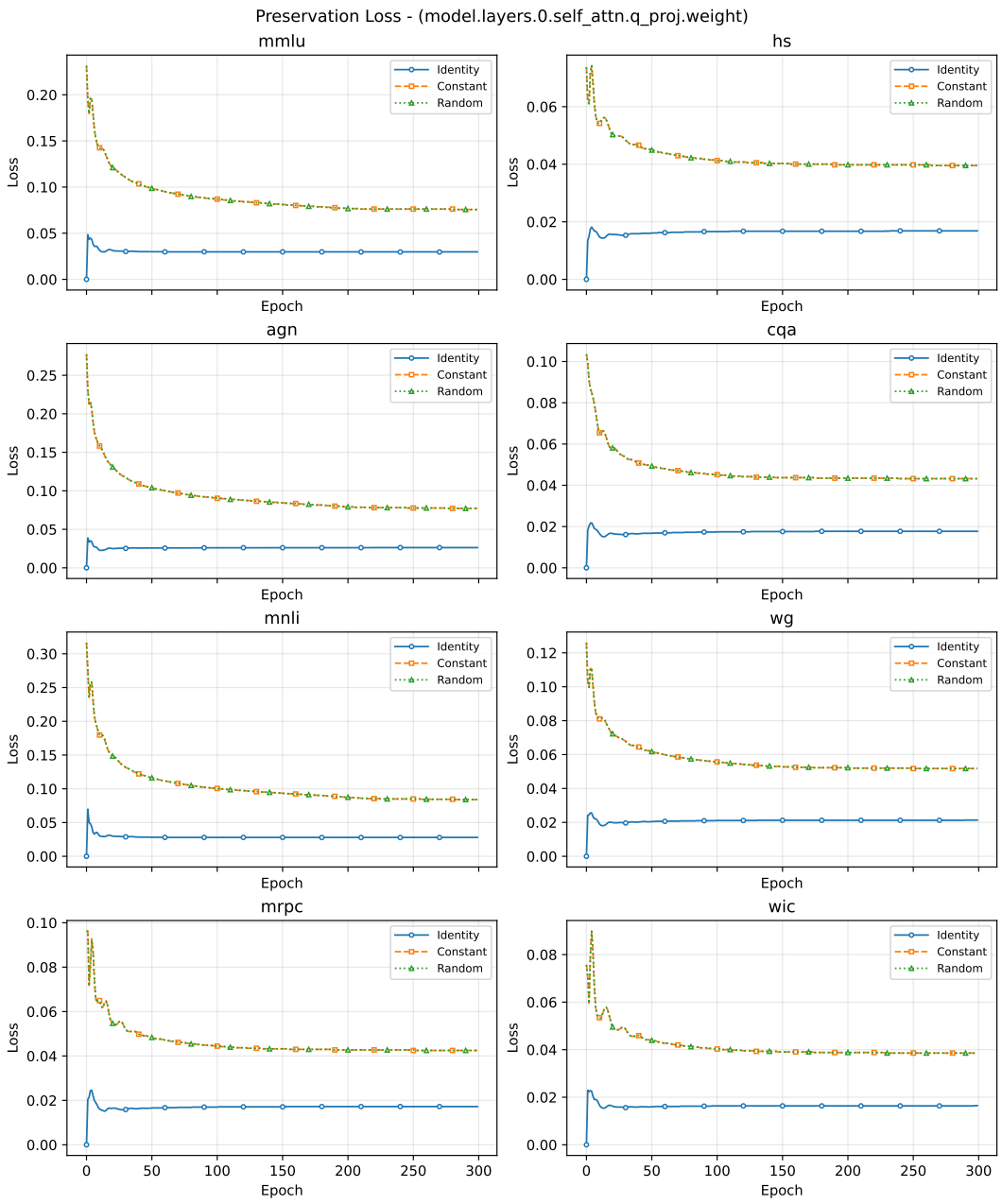


Figure 8: Preservation-loss trajectories for different gate initialization schemes on LLaMA-3.1-8B. Each panel corresponds to one task and reports the loss over 300 gate-update steps for `model.layers.0.self_attn.q_proj.weight`. Because $\boldsymbol{G}_t^{(0)} = \boldsymbol{I}$, Identity starts with zero preservation loss and generally remains below Constant and Random throughout optimization, indicating better preservation of the target task's surrogate input representation.

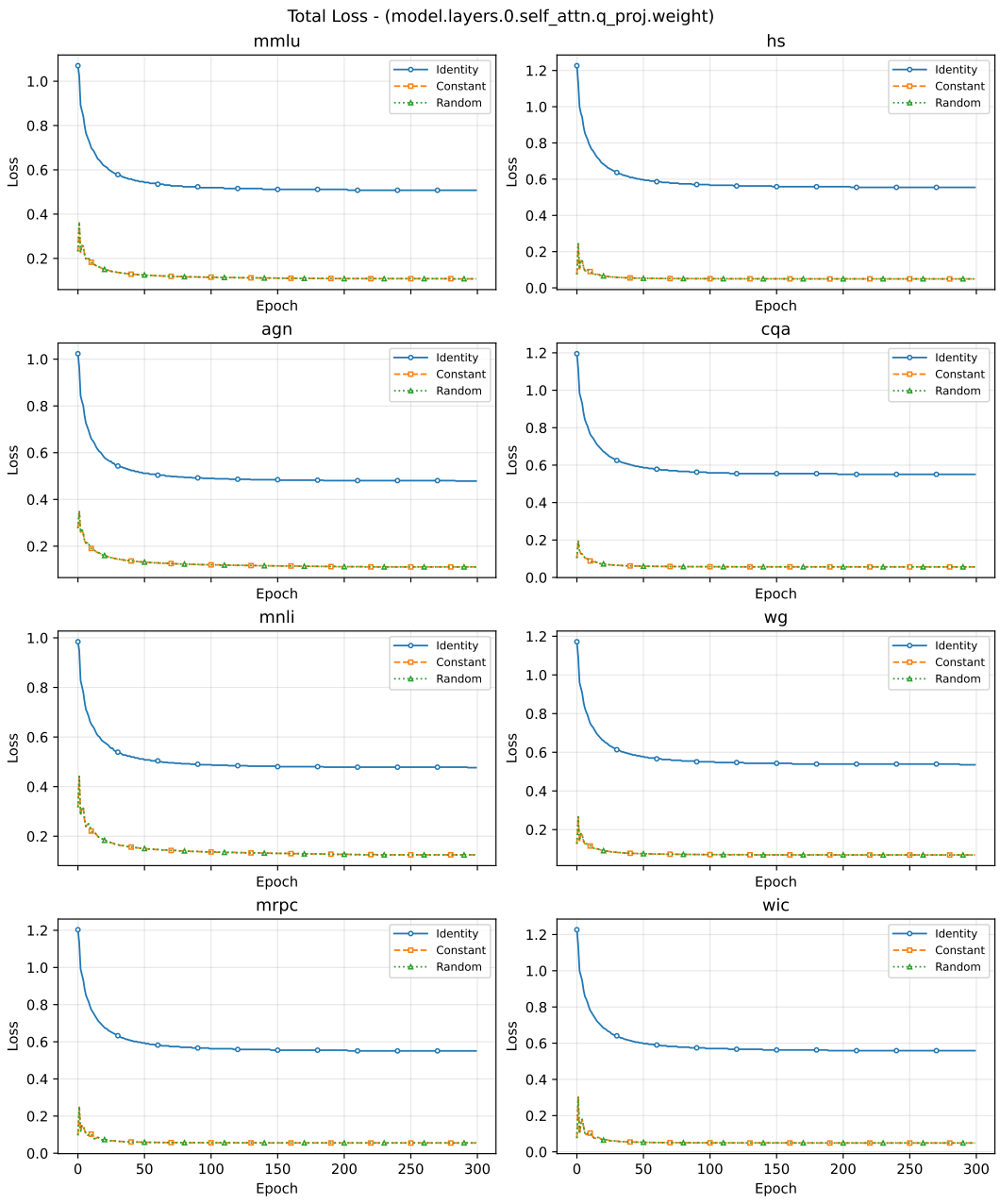


Figure 9: Total-loss trajectories for different gate initialization schemes on LLaMA-3.1-8B. The total loss is the sum of the suppression and preservation losses. Despite its lower preservation loss, Identity generally yields a higher total loss than Constant and Random because of its substantially larger suppression loss.

Method	MMLU	HS	AGN	CQA	MNLI	WG	MRPC	WiC	Avg.
Non-Merged									
Base	57.13	49.03	64.16	63.47	38.13	49.61	66.49	50.07	54.76
Fine-Tuned	69.02	96.05	95.53	82.31	92.08	88.62	88.70	75.64	85.99
Merged									
Identity	61.57	92.91	95.07	82.56	89.21	88.82	85.22	58.86	**81.78**
Constant	61.54	92.62	94.72	81.16	87.79	87.41	85.28	60.21	81.34
Random	61.50	92.61	94.70	81.24	87.75	87.35	84.99	60.00	81.27

Table 11: Downstream accuracy (%) after merging eight natural language tasks with LLaMA-3.1-8B under different gate initialization schemes. Base and Fine-Tuned report the pre-trained model and the individual task-specific models before merging, respectively. Identity, Constant, and Random use the same GTA optimization settings and differ only in gate initialization. The best average performance among the three initialization schemes is shown in bold.

Method	MMLU	HS	AGN	CQA	MNLI	WG	MRPC	WiC	Avg.
Non-Merged									
Base	27.65	25.03	31.32	21.29	35.45	49.42	66.49	50.00	38.33
Fine-Tuned	49.28	84.45	95.03	71.74	88.58	70.48	86.72	71.79	77.26
Merged									
Identity	43.96	74.73	90.79	66.42	59.62	68.92	75.01	57.29	**67.09**
Constant	44.66	74.94	90.42	65.77	55.34	67.46	73.51	56.29	66.05
Random	44.54	74.91	90.42	65.68	55.55	67.44	73.28	56.57	66.05

Table 12: Downstream accuracy (%) after merging eight natural language tasks with LLaMA-3.2-1B under different gate initialization schemes. Base and Fine-Tuned report the pre-trained model and the individual task-specific models before merging, respectively. Identity, Constant, and Random use the same GTA optimization settings and differ only in gate initialization. The best average performance among the three initialization schemes is shown in bold.

Figures 7–9, shown for LLaMA-3.1-8B, demonstrate that the initialization scheme changes the optimization trade-off between suppression and preservation. Identity produces a higher suppression loss but a lower preservation loss than Constant and Random. Consequently, Identity also exhibits a higher total loss and does not provide the lowest value of the aggregate surrogate objective. We observe the same qualitative ordering for LLaMA-3.2-1B, whose corresponding curves are omitted for concision. This behavior is qualitatively consistent with the finite-step perspective in Eq. 44: Identity begins with an exact preservation property, and the effect of this initial condition remains visible after a finite number of updates. Constant and Random, by contrast, reach lower values of the aggregate surrogate loss within the same optimization budget but do not retain the same preservation-oriented bias.

Nevertheless, Tables 11 and 12 show that Identity achieves the highest average merged-model accuracy at both model scales. On LLaMA-3.1-8B, Identity reaches 81.78%, compared with 81.34% for Constant and 81.27% for Random. On LLaMA-3.2-1B, it reaches 67.09%, compared with 66.05% for both alternatives. Thus, across both models, a lower total surrogate loss does not necessarily translate into better downstream performance. Theorem 2 characterizes how the identity-preserving residual arises, but it does not predict downstream accuracy. The results in Tables 11 and 12 therefore complement the theorem by showing that this residual can be beneficial for merging even when it yields a larger PSI surrogate loss. Overall, the analysis supports finite-step identity initialization as a preservation-oriented component of GTA rather than merely a numerical approximation to the quadratic optimum.