TADrop: Distribution-Guided Tensor-wise Adaptive Sparsification for Model Merging

Anonymous ACL submission

Abstract

Merging multiple task-specific fine-tuned checkpoints into a single multi-task model is an attractive alternative to joint multi-task training, especially when retraining or sharing all task data is infeasible. In many task-vector merging pipelines, each task vector is first sparsified before cross-task aggregation, removing low-magnitude updates that are often less informative and can accumulate as noise when combined with other tasks. However, existing sparsifiers often rely on a single manually tuned global sparsity ratio shared across all task vectors, which can over-sparsify some tensors while under-sparsifying others. To address this, we propose **TADrop** (**T**ensor-wise **A**daptive **D**rop), a simple plug-and-play sparsifier that automatically sets a tensor-specific sparsification ratio from a lightweight quantile statistic of the tensor's magnitude distribution, with an optional calibration step that uses only a small validation set. TADrop can be plugged into existing merging methods as a drop-in replacement for their sparsification stage, leaving the remaining merging pipeline unchanged. Across multiple merging methods and benchmarks spanning vision, language, and multimodal tasks, TADrop consistently improves merged performance, including a 4.99-point gain over EMR-Merging on RoBERTa GLUE and a 2.26-point gain in thirty-task ViT-B/16 merging. Further analyses suggest that TADrop better preserves task-relevant parameters while pruning redundant updates, thereby helping reduce interference during merging.

1 Introduction

Foundation models have become the default backbone for a wide range of applications, and adapting them to downstream needs is commonly done by fine-tuning and releasing task-specific checkpoints (Devlin et al., 2019; Dosovitskiy et al., 2021; Touvron et al., 2023). This practice has produced a rapidly expanding pool of specialized models, each tailored to a particular task, domain, or modality. Yet deploying these specialists at scale typically requires storing, serving, and routing a separate checkpoint for each task, which quickly becomes costly and difficult to maintain. Direct multi-task fine-tuning (Zhang and Yang, 2022) is one option, but it often requires joint access to all task data and substantial retraining, both of which can be impractical for foundation-scale models.

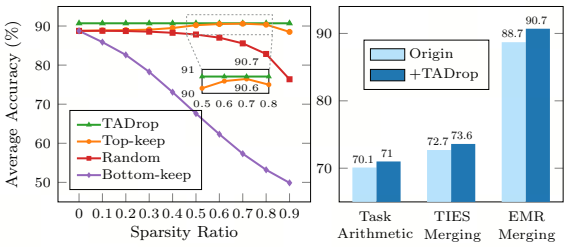
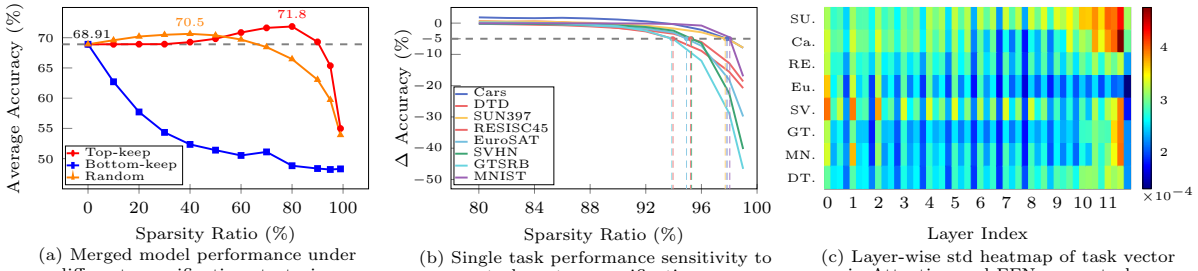


Figure 1: Global sparsity ratios require careful tuning, whereas TADrop provides consistent gains across representative merging methods. **Left**: Average accuracy over eight ViT-B/32 tasks after EMR Merging with different sparsification strategies and sparsity ratios. **Right**: Average accuracy before and after adding TADrop.

Model merging (Yang et al., 2024a; Wang et al., 2025) offers a more resource-friendly route by combining task-specific checkpoints directly in parameter space, often without additional training or shared access to task data. A widely adopted instantiation is the *task vector* formulation (Ilharco et al., 2023), which writes each fine-tuned model as a shared base plus a task-specific update and merges these updates. However, direct task-vector aggregation is prone to cross-task interference. Recent studies (Yadav et al., 2023; Yu et al., 2024) suggest that task vectors are highly uneven: a small fraction of large-magnitude entries carries most task-specific signal, while many low-magnitude entries are often less informative and can accumulate as noise when combined across tasks. This motivates sparsifying each task vector before cross-task aggregation, so that low-magnitude updates are fil-



(a) Merged model performance under different sparsification strategies.

(b) Single task performance sensitivity to task vector sparsification.

(c) Layer-wise std heatmap of task vector in Attention and FFN across tasks.

Figure 2: Task vectors are heterogeneous across sparsity levels, tasks, and layers, motivating tensor-wise adaptive sparsification. (a) illustrates how sparsification strategy and sparsity ratio affect merged-model performance; (b) shows that different tasks exhibit different sensitivity to task vector sparsification; (c) visualizes the heterogeneity of task vector magnitude distributions across layers/modules and tasks.

tered out before they interact with updates from other tasks.

However, existing sparsification practices for model merging are typically coarse and inflexible: most methods apply a single global sparsity ratio uniformly across task vectors (Yadav et al., 2023; Wang et al., 2024; Yang et al., 2024b). As shown in Figure 2a, merged accuracy varies substantially with the chosen global sparsity ratio, indicating that this hyperparameter is difficult to set reliably. Moreover, Figure 2b shows that different tasks tolerate sparsification to very different extents, suggesting that no single ratio can be uniformly optimal across task vectors. A shared ratio can therefore over-sparsify sensitive task vectors while under-sparsifying redundant ones, making fixed-ratio sparsification hard to tune and less portable. These limitations become especially problematic in large-scale multi-task settings, where task heterogeneity grows and interference is amplified.

To address these limitations, we propose **TADrop** (**T**ensor-wise **A**daptive **D**rop), a lightweight task-vector sparsification method that replaces a fixed global sparsity ratio with tensor-specific sparsification levels estimated from task-vector magnitude distributions. Rather than treating all tensors as equally sensitive to pruning, TADrop uses the distributional shape of each tensor as a proxy for its relative pruning tolerance: tensors with more prevalent large updates are more likely to carry task-relevant signal and are pruned conservatively, while tensors whose magnitudes concentrate near zero can often tolerate more aggressive pruning. We instantiate this principle with a rank-based quantile ratio, which is invariant to per-tensor rescaling and converts each tensor's magnitude distribution into a tensor-specific sparsification level. When a small validation set is available, we provide an optional module-level calibration step to further refine the resulting sparsification levels. TADrop is plug-and-play and method-agnostic: it can be applied either as a preprocessing step before merging or as a drop-in replacement for an existing fixed-ratio sparsification component, with no other changes to the pipeline. Across three representative merging methods (Ilharco et al., 2023; Yadav et al., 2023; Huang et al., 2024) and benchmarks spanning vision, language, and multimodal tasks, TADrop consistently improves merged performance. As illustrated in Figure 1, adding TADrop improves all three host merging methods on the eight ViT-B/32 tasks, including a 2-point gain on top of the strong EMR-Merging baseline.

Overall, this work makes three contributions. First, we identify fixed-ratio sparsification as a practical bottleneck in task-vector model merging, showing that sparsification sensitivity varies substantially across tasks and tensors. Second, we introduce TADrop, a distribution-guided tensor-wise sparsification method that adaptively assigns sparsification levels from task-vector magnitude distributions, with optional module-level calibration when a small validation set is available. Third, we validate TADrop across multiple merging methods and diverse vision, language, and multimodal benchmarks, demonstrating consistent gains and particularly strong improvements in many-task regimes.

2 Methodology

2.1 Preliminaries and Motivation

Let $\theta_{\text{pre}} = \{\theta^1_{\text{pre}}, \ldots, \theta^n_{\text{pre}}\}$ denote a pretrained base model consisting of n parameter tensors. Starting from the same initialization, we independently fine-tune it on K downstream tasks, producing K task-specific checkpoints $\{\theta_k\}_{k=1}^{K}$. Model merging

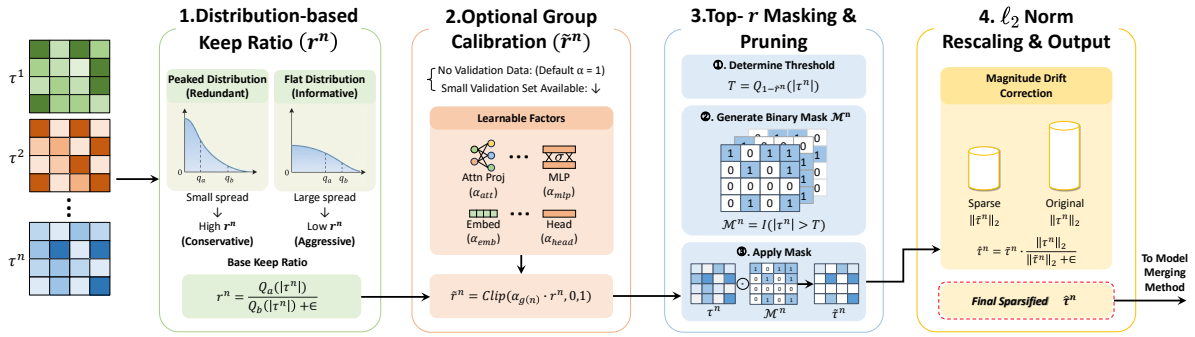


Figure 3: TADrop converts each task-vector tensor into a tensor-specific sparsified update before the host merging method is applied. Given task vectors $\{\tau^n\}$, TADrop estimates a keep ratio r^n from each tensor's magnitude distribution using a quantile ratio (Step 1), optionally calibrates it by functional module group when a small validation set is available (Step 2), applies Top-$\tilde{r}^n$ masking (Step 3), and preserves the tensor's ℓ_2 norm after masking (Step 4). The resulting sparsified task vectors can be passed to Task Arithmetic, TIES-Merging, EMR-Merging, or other downstream merging pipelines.

aims to construct a single multi-task model θ_{MTL} by combining these checkpoints, typically with limited or no access to the original task training data.

A prevalent paradigm in recent model merging is to merge in the *task vector* space (Ilharco et al., 2023). For each task k, the task vector is defined as the parameter delta relative to the pretrained model, $\tau_k = \theta_k - \theta_{pre}$, which captures task-specific updates induced by fine-tuning. Many merging methods can be written in the following unified form:

$$\theta_{\text{MTL}} = \theta_{\text{pre}} + \sum_{k=1}^{K} \lambda_k \, \Phi(\tau_k), \tag{1}$$

where λ_k controls each task's contribution (e.g., uniform or tuned weights), and $\Phi(\cdot)$ denotes operating on task vectors to mitigate interference before aggregation. In this work, we focus on the sparsification component of $\Phi(\cdot)$: before cross-task aggregation, each task vector τ_k is pruned independently according to its own magnitude distribution.

Empirical effects of sparsification. To examine the role of sparsification in merging, we apply a shared global ratio s to each task vector under three rules, *Top-keep*, *Bottom-keep*, and *Random*, and merge the result with a standard pipeline (Ilharco et al., 2023). Figure 2a shows that the merged accuracy is highly sensitive to the global ratio: *Top-keep* eventually dominates, improving the dense baseline from 68.9% to 71.8% at ∼80% sparsity, whereas *Bottom-keep* consistently degrades. This suggests that a small set of large-magnitude entries carries most of the task-specific signal while many small-magnitude entries act as noise during aggregation.

Limitation of a single global ratio. Although magnitude-based sparsification is effective, it leaves a key practical question: how much of each task vector should be retained? As shown in Figure 2b, different tasks tolerate sparsification to very different extents, suggesting that a single global sparsity ratio is difficult to set reliably across task vectors. Moreover, task-vector magnitude dispersion varies markedly across layers and modules (Figure 2c), indicating that a shared ratio can over-prune sensitive parts of task vectors while under-pruning redundant ones. This motivates an automatic, finer-grained sparsifier that adapts to the internal structure of each task vector rather than imposing a uniform sparsity level.

2.2 TADrop

The core intuition is that the *shape* of a task vector's magnitude distribution reflects how concentrated its task-specific updates are. Tensors whose magnitudes are concentrated near zero with only a few salient updates can be pruned aggressively, whereas tensors with more prevalent large-magnitude updates should be pruned more conservatively.

Distribution-guided tensor-wise sparsification. We quantify the distributional shape of each tensor using a scale-invariant ratio of two quantiles of $|\tau^n|$, which serves as a proxy for the tensor's keep ratio [1]:

$$r^n = \frac{Q_a(|\tau^n|)}{Q_b(|\tau^n|) + \epsilon}, \tag{2}$$

where $Q_p(\cdot)$ denotes the p-th quantile, $0 < a < b < 1$, and ϵ is a small constant. Because the

[1] We omit the task index k in what follows; all computations are applied independently to each τ_k^n.

ratio depends on relative ranks rather than absolute magnitudes, it is invariant to per-tensor rescaling and captures the shape of the magnitude distribution rather than the raw update scale. When large-magnitude entries are more prevalent, Q_a approaches Q_b, yielding a larger r^n and thus more conservative pruning; when such entries are rare, r^n becomes smaller and TADrop assigns a more aggressive sparsification level. A formal justification is provided in Appendix Section C.

Optional module-level calibration. TADrop can be used in two modes depending on whether validation examples are available. In the data-free mode, the quantile-based keep ratios are used directly. When a small validation set is available, we provide an optional calibration step to further refine these ratios at the level of coarse functional groups $\mathcal{G} = \{g_1, \ldots, g_m\}$, such as attention projections, MLP projections, embeddings, and output heads. For each group g, a scalar α_g rescales the per-tensor proxy of every tensor in that group:

$$\tilde{r}^n = \mathrm{Clip}\big(\alpha_{g(n)}\, r^n,\, 0,\, 1\big)\,, \tag{3}$$

where $g(n)$ denotes the group assignment of tensor n, and $\mathrm{Clip}(\cdot, 0, 1)$ ensures a valid keep ratio. The factor α_g adjusts the sparsification strength of group g, allowing calibration to favor more conservative or more aggressive pruning for different module types.

Masking and norm-preserving rescaling. Given the final keep ratio $\tilde{r}^n \in (0, 1)$, whose corresponding sparsity ratio is $1 - \tilde{r}^n$, we construct a binary mask by keeping the top-$\tilde{r}^n$ fraction of elements by magnitude:

$$\mathcal{M}^n = \mathbb{I}(|\tau^n| > Q_{1-\tilde{r}^n}(|\tau^n|))\,. \tag{4}$$

We then sparsify the tensor as

$$\tilde{\tau}^n = \tau^n \odot \mathcal{M}^n. \tag{5}$$

To mitigate magnitude drift introduced by masking, we rescale each sparsified tensor to preserve its ℓ_2 norm:

$$\hat{\tau}^n = \tilde{\tau}^n \cdot \frac{\|\tau^n\|_2}{\|\tilde{\tau}^n\|_2 + \epsilon}. \tag{6}$$

The rescaled tensor $\hat{\tau}^n$ replaces the original task-vector tensor in the host merging pipeline.

3 Experiments and Results

3.1 Setup

Models and datasets. Following common evaluation setups (Ilharco et al., 2023; Huang et al., 2024), we evaluate TADrop across vision, language, and multimodal settings. For **vision**, we use CLIP-based ViT-B/32 and ViT-L/14 encoders (Radford et al., 2021) on eight image classification tasks. For **language**, we use RoBERTa (Liu et al., 2019) on standard GLUE tasks as the main text benchmark, and report additional Llama-2 (Touvron et al., 2023) results in Section B.2 to cover autoregressive and larger-model settings. For **multimodal**, we build on BEiT3-base (Wang et al., 2023) and evaluate on five vision-language tasks, including generative tasks.

Baselines. We integrate TADrop into three representative merging methods, covering linear, magnitude-based, and task-modulated strategies: Task Arithmetic (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), and EMR-Merging (Huang et al., 2024). We also compare against TALL-masks (Wang et al., 2024), subspace-based ImPart (Yang et al., 2025a), and classical Weight Averaging, Fisher Merging, and RegMean baselines.

Implementation details. We set $a = 0.50$ and $b = 0.95$ by default. We report two variants: TADrop(w/o), which directly uses the quantile-based sparsification levels without validation data, and TADrop, which further applies optional module-level calibration with a small validation set. For calibration, we tune the module-level factors $\{\alpha_g\}$ via Optuna (Akiba et al., 2019) on a tiny calibration set (32 samples per task); for TADrop(w/o), we set $\alpha_g = 1$ for all groups. For each host merging method, we keep its original merging coefficients λ_k and other hyperparameters unchanged unless otherwise specified, so that TADrop only modifies the sparsification step. When combined with a host method, TADrop replaces or augments its sparsification stage while leaving the rest of the pipeline unchanged: it sparsifies each task vector before aggregation in Task Arithmetic, replaces the magnitude-pruning step in TIES-Merging while keeping sign alignment, and produces an adaptive mask combined with the consensus sign mask via element-wise multiplication in EMR-Merging.

Methods	SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD	Avg.
Individual	79.2 / 84.9	77.7 / 92.4	96.1 / 97.4	99.7 / 99.7	97.5 / 98.1	98.7 / 99.2	99.7 / 99.8	79.4 / 84.1	91.0 / 94.4
Traditional MTL	73.9 / 80.8	74.4 / 90.6	93.9 / 96.3	98.2 / 96.3	95.8 / 97.6	98.9 / 99.1	99.5 / 99.6	77.9 / 84.4	88.9 / 93.5
Weight Averaging	65.3 / 72.1	63.4 / 81.6	71.4 / 82.6	71.7 / 91.9	64.2 / 78.2	52.8 / 70.7	87.5 / 97.1	50.1 / 62.8	65.8 / 79.6
Fisher Merging	68.6 / 69.2	69.2 / 88.6	70.7 / 87.5	66.4 / 93.5	72.9 / 80.6	51.1 / 74.8	87.9 / 93.3	59.9 / 70.0	68.3 / 82.2
RegMean	65.3 / 73.3	63.5 / 81.8	75.6 / 86.1	78.6 / 97.0	78.1 / 88.0	67.4 / 84.2	93.7 / 98.5	52.0 / 60.8	71.8 / 83.7
PCB-Merging	66.7 / 76.8	65.5 / 86.2	78.5 / 89.4	79.3 / 96.5	86.4 / 88.3	77.1 / 91	98.2 / 98.6	59.1 / 73.6	76.3 / 87.5
Localize-and-Stitch	67.2 / -	68.3 / -	81.8 / -	89.4 / -	87.9 / -	86.6 / -	94.8 / -	62.9 / -	79.9 / -
AdaMerging++	66.6 / 79.4	68.3 / 90.3	82.2 / 91.6	94.2 / 97.4	89.6 / 93.4	89.0 / 97.5	98.3 / 99.0	60.6 / 79.2	81.1 / 91.0
ProDistill	68.9 / 77.7	71.2 / 90.0	89.9 / 94.4	99.4 / 99.5	96.1 / 97.7	95.3 / 98.3	99.5 / 99.6	68.0 / 78.2	86.0 / 91.9
Task Arithmetic	55.1 / 74.1	54.9 / 82.1	66.7 / 86.7	77.2 / 93.8	80.2 / 87.9	**69.7** / 86.8	**97.3** / 98.9	50.1 / 65.6	68.9 / 84.5
+ ImPart	62.2 / 71.2	60.7 / 81.3	71.1 / 84.7	76.6 / 93	73.5 / 83.5	63.4 / 79.8	93.5 / 98.3	50.9 / 64.3	69.0 / 82.0
+ TALL-masks	**65.9** / 74.5	**64.1** / 83.9	**75.5** / 88.8	**80.4 / 95.6**	78.4 / 88.4	69.6 / 85.0	95.9 / 98.9	**55.1** / 67.3	**73.1** / 85.3
+ TADrop(w/o)	62.7 / 74.3	61.7 / 82.9	72.2 / 87.3	78.2 / 93.5	77.4 / 88.1	67.5 / 87.7	95.9 / 99.0	52.2 / 66.8	71.0 / 84.9
+ TADrop	63.6 / **75.0**	62.5 / **85.5**	73.5 / **89.0**	77.9 / 94.5	**80.7 / 91.0**	68.8 / **90.5**	96.7 / **99.2**	52.8 / **68.5**	72.1 / **86.7**
TIES-Merging	**64.9** / 74.4	64.3 / 84.5	74.6 / 88.0	76.4 / 94.0	81.2 / 85.7	69.3 / 82.1	96.5 / 98.7	54.2 / 67.7	72.7 / 84.4
+ ImPart	60.7 / 73.6	59.3 / 82.8	70.6 / 87.1	81.4 / 93.9	81.2 / 87.9	71.8 / 86.5	97.2 / 98.9	52.8 / 66.3	71.9 / 84.6
+ TALL-masks	66.8 / 74.4	**65.8** / 84.5	**75.6** / 87.7	77.6 / 94.1	80.6 / 85.2	69.7 / 81.5	96.2 / 98.6	**55.3** / 67.7	73.5 / 84.2
+ TADrop(w/o)	64.8 / **76.0**	63.0 / 83.6	74.1 / 88.5	**82.1** / 94.3	81.5 / 89.9	71.1 / 86.6	97.1 / 99.0	55.2 / 67.5	73.6 / 85.7
+ TADrop	64.4 / 75.8	63.4 / **85.2**	75.2 / **89.1**	80.7 / **94.9**	**85.5 / 91.3**	**74.4 / 90.2**	**97.9 / 99.2**	54.4 / **68.6**	**74.5 / 86.8**
EMR-Merging	75.2 / 83.2	72.8 / 90.7	93.5 / 96.8	**99.5 / 99.7**	96.9 / 97.9	98.1 / 99.1	99.6 / 99.7	74.4 / 82.7	88.7 / 93.7
+ TADrop(w/o)	**78.7** / 84.3	78.6 / 92.1	95.2 / **97.4**	99.4 / **99.7**	**97.4** / 98.1	98.7 / 99.1	**99.7 / 99.8**	77.9 / 83.5	90.7 / 94.2
+ TADrop	**78.7 / 84.4**	**79.0 / 92.2**	**95.4** / 97.2	**99.5 / 99.7**	**97.4 / 98.2**	**98.8 / 99.2**	99.6 / **99.8**	**78.2 / 84.0**	**90.8 / 94.4**

Table 1: Multi-task performance when merging ViT models on eight tasks (a / b: a = ViT-B/32, b = ViT-L/14). **Bold** numbers indicate the best score within each backbone block, and underlined numbers indicate the second-best score.

3.2 Results

Performance on ViT models. Table 1 summarizes multi-task merging results on eight vision classification tasks using ViT-B/32 and ViT-L/14 backbones. Across Task Arithmetic, TIES-Merging, and EMR-Merging, incorporating TADrop improves average performance on both backbones, demonstrating that tensor-wise adaptive sparsification is broadly effective across different merging pipelines. Notably, TADrop remains effective on the strong EMR-Merging baseline, improving the average accuracy by +2.1 and +0.7 points on ViT-B/32 and ViT-L/14, respectively. Compared with sparsification-related baselines, TADrop outperforms ImPart and achieves stronger average performance than TALL-masks on ViT-L/14, while remaining competitive on ViT-B/32. Finally, while the data-free variant TADrop(w/o) achieves most of the performance gain, applying module-level calibration can further improve accuracy by fine-tuning tensor-wise sparsification levels.

Performance on RoBERTa models. Table 2 reports average performance on eight GLUE text classification tasks with RoBERTa-base. Across merging pipelines, integrating TADrop consistently improves the corresponding merged models, with gains of +3.13, +5.60, and +4.99 points over Task Arithmetic, TIES-Merging, and EMR-Merging, respectively. These results suggest that tensor-wise adaptive sparsification is also effective for language model merging.

Method	Avg.
Individual	85.56
Weight Averaging	51.34
Fisher Merging (Matena and Raffel, 2022)	45.57
RegMean (Jin et al., 2023)	70.01
Task Arithmetic (Ilharco et al., 2023)	66.65
+ ImPart (Yang et al., 2025a)	65.72
+ TALL-masks (Wang et al., 2024)	54.55
+ TADrop(w/o)	68.35
+ TADrop	**69.78**
TIES-Merging (Yadav et al., 2023)	64.04
+ ImPart (Yang et al., 2025a)	61.45
+ TALL-masks (Wang et al., 2024)	66.45
+ TADrop(w/o)	67.51
+ TADrop	**69.64**
EMR-Merging (Huang et al., 2024)	80.18
+ TADrop(w/o)	84.94
+ TADrop	**85.17**

Table 2: Average multi-task performance when merging RoBERTa-base models fine-tuned on eight GLUE text classification tasks. **Bold** numbers indicate the best score within each backbone block, and underlined numbers indicate the second-best score.

Performance on BEiT3 models. Table 3 reports results on five vision-language tasks using BEiT3-base. This benchmark combines retrieval, classification, and captioning objectives, making it a more heterogeneous and challenging testbed for

Methods	COCO-Re	COCO-Captioning				ImageNet	NLVR2	VQAv2	Avg.
	Acc	BLEU	CIDEr×100	METEOR	ROUGE	Acc	Acc	Acc	Score
Individual	84.6	39.4	134.0	31.1	60.1	85.4	77.7	84.4	74.6
Weight Averaging	18.9	3.1	0.1	11.5	15.9	67.7	28.0	62.9	26.0
Task Arithmetic (Ilharco et al., 2023)	31.8	3.3	0.0	11.8	17.6	70.8	38.1	69.3	30.3
TIES-Merging (Yadav et al., 2023)	39.3	2.9	0.1	10.8	16.7	69.8	32.1	67.2	29.9
EMR-Merging (Huang et al., 2024)	79.5	28.9	106.0	27.2	53.4	77.4	74.8	72.1	64.9
+ TADrop(w/o)	80.5	32.4	**119.0**	**28.7**	56.0	80.1	77.3	72.1	68.3
+ TADrop	**80.9**	**33.7**	118.0	**28.7**	**56.2**	**80.5**	**77.4**	**72.4**	**68.5**

Table 3: Performance of merging multimodal BEiT3 models on five vision-language tasks. **Bold** numbers indicate the best score within each backbone block, and underlined numbers indicate the second-best score.

model merging. We focus on the strong EMR-Merging baseline in this setting, and TADrop further improves its average score from 64.9 to 68.5. For example, COCO captioning improves from 28.9 to 33.7 in BLEU and from 106.0 to 118.0 in CIDEr×100. These results indicate that adaptive task-vector sparsification also benefits multimodal merging, where task types and output formats vary substantially.

Data-free and calibrated variants. Across the evaluated settings, the data-free variant TADrop(w/o) already provides clear gains over the corresponding host methods, while module-level calibration usually brings additional improvements. This suggests that the distribution-based tensor-wise ratios capture most of the useful sparsification signal without requiring validation data, and the small calibration set mainly serves to adjust coarse module preferences when such data is available. For example, under EMR-Merging, TADrop(w/o) improves the ViT-B/32 average from 88.7 to 90.7 and RoBERTa from 80.18 to 84.94, while the calibrated variant further reaches 90.8 and 85.17, respectively. These results indicate that validation data is not required for TADrop to be effective, but can be used as an optional source of signal for further refinement.

4 Analyses

We next examine when and why TADrop improves merging. The analyses cover scalability with increasing task count, preservation of task-associated neurons, sensitivity to quantile choices, and the role of norm-preserving rescaling.

4.1 Scalability in Many-Task Merging

Merging more tasks typically increases cross-task interference, making uniform or naive sparsification less effective. To evaluate TADrop in such challenging scenarios, we progressively merge 8 to 30 ViT-B/16 tasks using EMR-Merging as the baseline. As shown in Figure 4, TADrop consistently improves average accuracy over EMR-Merging across all merge scales. The gain grows from 0.74 points with 8 tasks to 1.31 points with 24 tasks, reaching 2.26 points at 30 tasks. The increasing gain suggests that tensor-wise adaptivity becomes more useful as task heterogeneity accumulates, which is consistent with our motivation that fixed sparsification becomes less adequate in many-task regimes. Full 30-task results are provided in Table 6 in the Appendix.

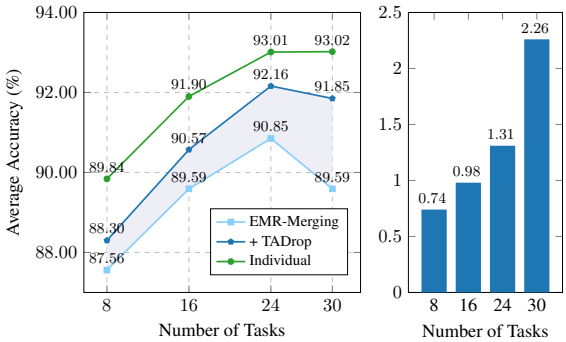


Figure 4: Scalability of TADrop in many-task merging on ViT-B/16. **Left**: Absolute average accuracy as the number of merged tasks grows from 8 to 30. **Right**: Performance gain over the EMR-Merging baseline.

4.2 Task Neurons Analysis

TADrop assigns adaptive keep ratios across tensors based on task-vector distributions. We therefore ask whether this adaptivity helps preserve task-relevant functional units. To probe this, we perform a neuron-level analysis on the MLP module, which is often viewed as storing task-specific features. We construct a unified probe set by pooling samples from all tasks and record neuron activations in the MLP expansion layers. For each neuron i, we collect the top-k samples that elicit the largest activation of i, and assign i a task label by majority vote over the task identities of these top-k samples (e.g., if 8 of the top-k samples are from Cars and 2

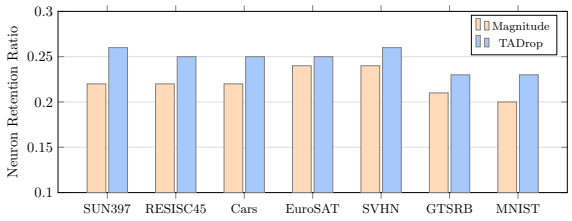


Figure 5: Retention ratio of task-labeled neurons. TADrop retains a higher fraction of task-labeled neurons than the magnitude-based baseline under the same overall sparsity.

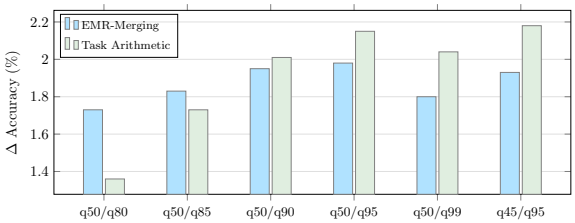


Figure 6: Ablation on quantile choices (Q_a, Q_b) on ViT-B (8 tasks). Bars show the change in average accuracy relative to each host method's baseline.

Method	EMR-Merging	w/o R	$(1-p)$ R	ℓ_2 R
Avg.	88.7	86.8	71.1	**90.8**

Table 4: Ablation results of rescaling strategies on the 8 ViT tasks. R denotes the rescaling operation.

from MNIST, neuron i is labeled as a Cars neuron). We then sparsify task vectors using (i) TADrop and (ii) a tensor-wise magnitude-based Top-k baseline; for fair comparison, the baseline follows the same tensor-wise masking granularity and matches the overall sparsity level induced by TADrop.

As shown in Figure 5, TADrop retains a higher fraction of task-labeled neurons than the tensor-wise magnitude baseline under the same overall sparsity. This trend is consistent across all evaluated vision tasks, suggesting that TADrop robustly preserves task-specific functional units in the MLP layers. The higher retention likely contributes to the improved merged accuracy, as critical task information is maintained while redundant entries are pruned. By adapting sparsification to the distributional shape of each tensor, TADrop selectively protects neurons with strong task-relevant signals, in contrast to a uniform magnitude-based rule that treats all tensors equivalently.

4.3 Ablation Study

Quantile selection. TADrop uses two quantile levels (a, b) to estimate tensor-wise sparsification levels from the magnitude distribution of each task vector. A natural question is whether the method is sensitive to this choice, since task-vector distributions may vary across tasks and models. We therefore ablate several quantile pairs on ViT-B over eight vision tasks by integrating TADrop into two baselines, EMR-Merging and Task Arithmetic, and report the change in average accuracy relative to each host method.

As shown in Figure 6, TADrop improves both pipelines across a broad range of quantile pairs. This indicates that the effectiveness of TADrop does not rely on a finely tuned quantile configuration; rather, the quantile ratio serves as a robust distributional proxy for assigning relative sparsification levels across tensors. Among the tested settings, $(a, b) = (0.50, 0.95)$ yields consistently strong gains across both pipelines, and we use it as the default in all experiments.

Rescaling strategy. We next study the effect of the rescaling step applied after masking, which is intended to mitigate magnitude drift introduced by sparsification before aggregation. We compare three variants on ViT-B with 8 vision tasks using EMR-Merging: (i) *w/o* R, which applies masking without any rescaling; (ii) $(1-p)$ R, multiplying the remaining entries by a constant factor $1/(1-p)$, where p is the overall drop rate, following the previously used with random dropping (Yu et al., 2024); and (iii) ℓ_2 R, our tensor-wise norm-preserving rescaling that matches the ℓ_2 norm of each tensor before and after masking.

Section 4.3 shows that the choice of rescaling substantially affects performance. Removing rescaling reduces average accuracy from 88.7 to 86.8, indicating that masking alone can introduce magnitude drift that harms subsequent aggregation. The $(1-p)$ correction performs poorly in our setting, likely because it is designed for random dropping, whereas magnitude-based pruning selectively retains large entries; under high drop rates, $1/(1-p)$ can also become large and over-amplify the retained updates. In contrast, ℓ_2 rescaling yields the best result, suggesting that preserving per-tensor norm provides a more stable correction for sparsified task vectors.

4.4 Visualization of Tensor-wise Sparsity

To better interpret the masks produced by TADrop, we visualize the tensor-wise sparsity ratios assigned across layers on a representative task (SUN397). As shown in Figure 7, the induced sparsity varies across parameter groups and layers,

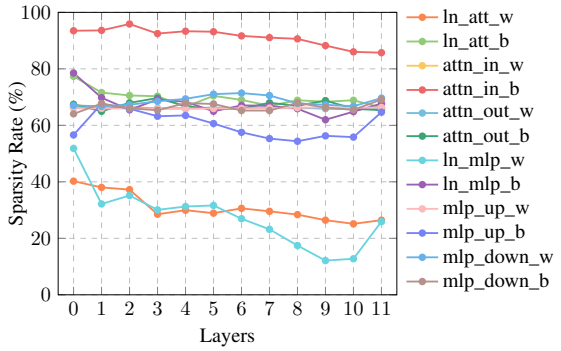


Figure 7: Tensor-wise sparsity ratios assigned by TADrop across layers.

reflecting that TADrop allocates different pruning strengths to different tensors rather than using a single global ratio. This visualization provides a qualitative view of TADrop's behavior and is consistent with the intended design of adapting sparsification according to per-tensor magnitude distribution characteristics.

5 Related Work

Task-vector based model merging has emerged as a practical approach to consolidate task-specific fine-tuned models without requiring joint access to all datasets. A key challenge in merging is **cross-task interference**, as naively summing task vectors often leads to destructive interactions. Consequently, many methods focus on sparsification or related conflict-mitigation strategies to preserve task-specific signals during aggregation.

Early approaches, such as DARE (Yu et al., 2024) and TIES-Merging (Yadav et al., 2023), rely on magnitude-based heuristics, including random dropping, threshold pruning, and sign-consensus, to reduce interference. Subsequent work explored refinements to these heuristics, e.g., magnitude-informed sampling (Deep et al., 2024) or conflict-aware balancing across layers (Yang et al., 2025b), highlighting the importance of aligning pruning strategies with the heterogeneous structure of task vectors. EMR-Merging (Huang et al., 2024) further demonstrated that careful combination of shared and task-specific modulation can achieve strong performance. Less is More (Qi et al., 2025) explores a complementary direction by binarizing task vectors into lightweight task switches, emphasizing efficiency and modular composition.

Orthogonal to magnitude-based heuristics, structure-aware and subspace methods attempt to exploit inherent organization in neural parameters. TALL-masks (Wang et al., 2024) and Localize-and-Stitch (He et al., 2024) identify task-specific subsets of parameters, while Twin-Merging (Lu et al., 2024) dynamically integrates shared and task-specific modules at inference. NeuronMerge (Gu et al., 2025) leverages functional groupings of neurons, particularly in MLP layers, to guide merging. Low-rank or subspace-based methods, including ImPart (Yang et al., 2025a), SVD-based selection (Lee et al., 2025), and knowledge-aware subspace purification (An et al., 2025), further aim to reduce the effective dimensionality of task updates to improve both efficiency and robustness. Optimization-based formulations, such as adaptive projective gradient descent (Wei et al., 2025), represent another line of work that treats merging as a constrained optimization problem over task vectors.

Across these directions, a common limitation is the reliance on coarse, manually chosen global sparsity ratios, which do not account for tensor-level heterogeneity or task-specific distributional characteristics. TADrop proposes a tensor-wise adaptive sparsification strategy based on quantile statistics of task-vector magnitude distributions, with an optional lightweight module-level calibration step when a small validation set is available. This design provides a mechanism to modulate pruning according to tensor-level statistics, preserving task-relevant structure while remaining compatible with existing merging pipelines. For a broader conceptual comparison of sparsification strategies, see Appendix Section A.

6 Conclusion

In this paper, we presented TADrop, a method-agnostic sparsification approach for task-vector model merging. TADrop replaces a manually chosen global sparsity ratio with tensor-wise sparsification levels guided by the magnitude distribution of each task-vector tensor, with optional module-level calibration when a small validation set is available. This distribution-guided design allows sparsification to reflect tensor-level heterogeneity, rather than imposing the same pruning strength across all task vectors. Across vision, language, and multimodal benchmarks, TADrop consistently improves existing merging pipelines, with particularly clear gains in many-task merging scenarios. These results suggest that exploiting the distributional structure of task vectors is a practical way to reduce interference in model merging.

Limitations

While TADrop demonstrates consistent gains across a range of vision, language, and multimodal benchmarks, several limitations remain. First, our experiments focus on task vectors derived from shared pretrained backbones; the behavior of TADrop on more heterogeneous model families or alternative fine-tuning strategies, such as parameter-efficient adapters, remains to be explored. Second, the optional module-level calibration currently relies on coarse architectural groupings; automated or more fine-grained adaptation mechanisms may further enhance performance. Third, while TADrop scales to dozens of tasks in our experiments, its behavior on extremely large-scale merging scenarios (e.g., very large LLMs) has not been extensively evaluated. Exploring these aspects could provide a deeper understanding of the method's behavior and inform potential extensions in more complex or heterogeneous settings.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Bang An, Yibo Yang, Philip Torr, and Bernard Ghanem. 2025. Purifying task vectors in knowledge-aware subspace for model merging. *arXiv preprint arXiv:2510.14697*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Mark Chen. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Hetarth Chopra, Vidhi Rambhia, and Vikram Adve. 2025. Lewis (layer wise sparsity)–a training free guided model merging approach. *arXiv preprint arXiv:2503.03874*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim Kuan Goh, Ho-Kin Tang, Daojing He, and Min Zhang. 2024. Parameter competition balancing for model merging. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Wangyun Gu, Qianghua Gao, Zhang Li-Xin, Xu Shen, and Jieping Ye. 2025. Neuronmerge: Merging models via functional neuron groups. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9015–9037.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. 2024. Emr-merging: Tuning-free high-performance model merging. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Chanhyuk Lee, Jiho Choi, Chanryeol Lee, Donggyun Kim, and Seunghoon Hong. 2025. Adarank: Adaptive rank pruning for enhanced model merging. *CoRR*, abs/2503.22178.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. 2024. Twin-merging: Dynamic integration of modular expertise in model merging. *Advances in Neural Information Processing Systems*, 37:78905–78935.

Michael Matena and Colin Raffel. 2022. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Biqing Qi, Fangyuan Li, Zhen Wang, Junqi Gao, Dong Li, Peng Ye, and Bowen Zhou. 2025. Less is more: Efficient model merging with binary task switch. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15265–15274.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pages 8748–8763.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jiménez, François Fleuret, and Pascal Frossard. 2024. Localizing task information for improved model merging and compression. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. 2023. Image as a foreign language: BEIT pretraining for vision and vision-language tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 19175–19186.

Zijing Wang, Yongkang Liu, Yingfeng Luo, Ming Wang, Zhen Song, Shi Feng, Xiaocui Yang, Dingyang Lin, Daling Wang, Yifei Zhang, and 1 others. 2025. Scaling intelligence through model merging: A comprehensive survey. *Authorea Preprints*.

Yongxian Wei, Anke Tang, Li Shen, Zixuan Hu, Chun Yuan, and Xiaochun Cao. 2025. Modeling multi-task model merging as adaptive projective gradient descent. *arXiv preprint arXiv:2501.01230*.

Jun Wu, Jiangtao Wen, and Yuxing Han. 2025a. Backslash: Rate constrained optimized training of large language models. *Preprint*, arXiv:2504.16968.

Jun Wu, Yirong Xiong, Jiangtao Wen, and Yuxing Han. 2025b. It takes a good model to train a good model: Generalized gaussian priors for optimized llms. *CoRR*, abs/2506.00486.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024a. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *Preprint*, arXiv:2408.07666.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024b. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Yan Yang, Yixia Li, Hongru Wang, Xuetao Wei, James Jianqiao Yu, Yun Chen, and Guanhua Chen. 2025a. Impart: Importance-aware delta-sparsification for improved model compression and merging in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 18817–18829.

Zongzhen Yang, Binhang Qi, Hailong Sun, Wenrui Long, Ruobing Zhao, and Xiang Gao. 2025b. CABS: conflict-aware and balanced sparsification for enhancing model merging. *CoRR*, abs/2503.01874.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine*

Learning, ICML 2024, Vienna, Austria, July 21-27, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Yu Zhang and Qiang Yang. 2022. A survey on multi-task learning. *IEEE Trans. Knowl. Data Eng.*, 34(12):5586–5609.

A Conceptual Comparison of Sparsification Methods

Table 5 provides a non-exhaustive conceptual summary of representative sparsification strategies used in or related to model merging, grouped by their core pruning signal, data requirement, and adaptivity granularity. We include this table to help clarify the different assumptions behind these approaches and to situate TADrop within the broader design space.

B More Evaluation Results

B.1 Results for 30 ViT Tasks

Table 6 reports detailed results for merging 30 ViT-B/16 task models. Task Arithmetic and TIES-Merging obtain average accuracies of 48.89 and 37.53, respectively, whereas EMR-Merging remains strong at 89.59. Adding TADrop to EMR-Merging further increases the average accuracy to 91.85, showing that tensor-wise adaptive sparsification continues to provide gains at larger merge scales.

B.2 Results on Llama-2

We further evaluate whether TADrop transfers to large language models by testing on Llama-2 (Touvron et al., 2023) across representative chat/knowledge, math, and code benchmarks, including MMLU (Hendrycks et al., 2020), HellaSwag (Zellers et al., 2019), TruthfulQA (Lin et al., 2022), GSM8K (Cobbe et al., 2021), HumanEval (Chen, 2021), and MBPP (Austin et al., 2021). Due to limited compute, we report results on a fixed subset of 100 test examples per benchmark (the same subset for all methods). We integrate TADrop into two representative merging methods, TIES-Merging and EMR-Merging.

Table 7 summarizes the results under this subset setting. Across both methods, adding TADrop generally improves or matches the corresponding baseline on most benchmarks, with clearer gains on knowledge-style and math evaluations (e.g., MMLU and GSM8K), while maintaining comparable performance on code benchmarks (HumanEval/MBPP). These results suggest that tensor-wise adaptive sparsification can be beneficial beyond encoder-style models, including in LLM merging.

C Theoretical Justification

This section provides theoretical insight into the design of TADrop. We show that the quantile ratio used in our method is (i) invariant to the scale of a tensor and (ii) sensitive to the distributional shape that controls how concentrated the magnitudes are. These properties justify using the ratio as an effective proxy for setting tensor-wise keep ratios.

Empirical evidence suggests that parameter updates in neural networks often exhibit non-Gaussian, heavy-tailed behavior. Motivated by prior analyses that model neural network parameters/updates with generalized Gaussian families(Wu et al., 2025b,a), we use the **Generalized Gaussian Distribution (GGD)** as a convenient parametric model for task-vector entries.

Let X denote a signed coefficient in a task vector tensor and assume $X \sim \mathrm{GGD}(0, \alpha, \beta)$ with density

$$f_X(x; \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left(-\left(\frac{|x|}{\alpha}\right)^{\beta}\right), \tag{7}$$

where $\alpha > 0$ is a scale parameter and $\beta > 0$ controls peakedness and tail behavior. We focus on the magnitude $Y = |X|$, since TADrop operates on $|\tau|$.

TADrop uses a quantile ratio $R = Q_a/Q_b$ of the magnitude distribution to guide tensor-wise retention. Let $Q(p)$ denote the quantile function of $Y = |X|$, i.e., $\Pr(Y \le Q(p)) = p$. For the GGD, the CDF of Y satisfies

$$\Pr(Y \le y) = P\left(\frac{1}{\beta}, \left(\frac{y}{\alpha}\right)^{\beta}\right), \tag{8}$$

where $P(\cdot, \cdot)$ is the regularized lower incomplete gamma function. Solving for the quantile yields

$$Q(p; \alpha, \beta) = \alpha \cdot \left[P^{-1}\left(\frac{1}{\beta}, p\right)\right]^{1/\beta}, \tag{9}$$

where P^{-1} is the inverse in the second argument.

Category	Method	Core sparsification signal	Data	Adaptivity
Ours	TADrop	Tensor-wise pruning guided by quantile statistics of task-vector magnitudes	Optional	Tensor-wise
Magnitude-based	TIES-Merging (Yadav et al., 2023)	Top-k magnitude pruning	None	Global
	DELLA-Merging (Deep et al., 2024)	Magnitude-informed sampling and pruning	None	Global
	DARE (Yu et al., 2024)	Random pruning with rescaling	None	Global
Structure-aware	ImPart (Yang et al., 2025a)	SVD-based pruning of spectral components	None	Spectral
	CABS / PCB (Du et al., 2024; Yang et al., 2025b)	Pruning based on inter-task conflict or competition	None	Parameter / layer
Data-guided	LEWIS (Chopra et al., 2025)	Layer-wise pruning based on activation norms	Required	Layer-wise
	AdaRank (Lee et al., 2025)	Calibration-based sparse mask or rank selection	Required	Spectral

Table 5: Conceptual comparison of representative sparsification methods for model merging, grouped by their pruning signal, data requirement, and adaptivity level.

Define the quantile ratio for $0 < a < b < 1$:

$$R(a,b) = \frac{Q(a;\alpha,\beta)}{Q(b;\alpha,\beta)} = \left(\frac{P^{-1}(1/\beta,a)}{P^{-1}(1/\beta,b)}\right)^{1/\beta}. \tag{10}$$

Property 1: Scale invariance. The scale α cancels in $R(a,b)$, so the ratio is invariant to rescaling of a tensor. This supports comparing tensors with different absolute update magnitudes using a common proxy.

Property 2: Sensitivity to distributional shape. Eq. (8) shows that $R(a,b)$ depends only on the shape parameter β. Intuitively, as the distribution becomes flatter (larger β), lower and upper quantiles become closer relative to each other, increasing R; when the distribution is more peaked with heavier tails (smaller β), the separation between quantiles becomes larger, decreasing R. In the limiting case $\beta \to \infty$, X approaches a uniform distribution on $[-\alpha,\alpha]$ and thus $Y \sim \mathrm{Uniform}(0,\alpha)$, giving $R(a,b) \to a/b$. In practice, $R(a,b)$ varies smoothly with β and provides a useful proxy for magnitude concentration.

Motivated by this behavior, TADrop uses a monotone mapping from the shape proxy $R(a,b)$ to a tensor-wise keep ratio r, encouraging more aggressive pruning for highly concentrated tensors and more conservative retention for diffuse tensors.

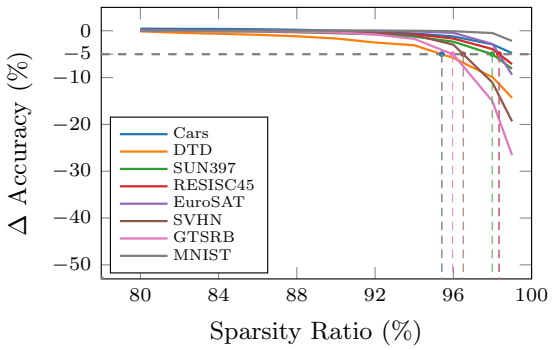


Figure 8: Single task performance sensitivity to task vector sparsification. (ViT-L/14).

	Pet	Cifar-10	Cars	MNIST	Kvasir-v2	Food-101	Weather	EuroSAT	Vegetables	Cats and Dogs
Individual	92.23	97.88	85.06	99.22	94.31	87.94	98.21	99.11	100.00	99.03
Weight Averaging	31.26	42.91	7.74	27.63	25.27	68.02	61.06	24.37	83.20	91.28
RegMean	34.62	89.65	16.28	90.71	71.00	76.14	86.62	74.13	99.10	98.54
Task Arithmetic	33.24	59.86	9.34	30.81	31.05	73.06	74.56	31.25	91.97	93.61
Ties-Merging	12.84	42.82	5.30	23.21	21.09	73.22	72.86	10.98	92.31	91.88
AdaMerging	48.34	87.54	0.42	81.22	22.76	75.23	89.13	44.60	97.97	96.91
EMR-Merging	86.50	96.73	43.50	97.79	84.07	83.22	96.97	**97.85**	99.96	**99.49**
+TADrop	**90.22**	**97.71**	**70.74**	**98.77**	**92.39**	**87.60**	**97.72**	97.11	**100.00**	99.44
	DTD	Fashion	Dogs	STL-10	Flowers	LandScape	RESISC45	EMNIST	Intel-Images	CUB-200-2011
Individual	71.75	93.26	89.90	99.07	98.19	94.00	98.90	94.67	94.63	91.88
Weight Averaging	14.63	20.46	47.80	37.74	68.97	73.14	13.56	7.73	82.40	37.66
RegMean	30.53	83.42	42.89	78.94	95.26	83.64	34.66	48.67	93.60	49.78
Task Arithmetic	14.73	37.11	47.65	39.66	80.68	79.59	15.50	11.05	89.03	41.86
Ties-Merging	3.71	27.05	26.03	6.17	34.33	78.27	6.79	5.61	89.39	31.28
AdaMerging	16.68	76.76	53.09	68.91	95.69	81.98	24.83	18.02	91.02	48.19
EMR-Merging	59.62	89.77	86.48	98.23	97.91	94.60	**94.80**	93.07	95.46	82.86
+TADrop	**61.22**	**91.64**	**89.70**	**99.04**	**97.94**	**95.00**	93.87	**93.73**	**95.50**	**84.98**
	SVHN	Cifar-100	Beans	SUN397	Garbage	Animal-10N	Fruits-360	GTSRB	MangoLeafBD	KenyanFood13
Individual	96.21	89.85	96.87	87.51	98.83	92.52	100.00	95.74	100.00	82.57
Weight Averaging	10.88	77.98	70.98	57.42	22.89	46.00	5.38	15.00	68.58	33.55
RegMean	66.13	82.59	92.58	58.58	65.31	68.74	19.79	56.96	98.10	57.11
Task Arithmetic	17.41	80.20	84.62	55.88	25.23	51.14	6.15	37.01	87.02	36.32
Ties-Merging	10.54	78.61	67.22	52.69	3.91	19.13	1.50	40.74	76.58	19.90
AdaMerging	25.70	84.19	93.38	64.09	38.54	66.55	7.94	59.90	99.13	48.66
EMR-Merging	90.56	87.80	95.31	80.92	**94.43**	88.68	97.40	95.72	99.97	78.52
+TADrop	**91.94**	**90.44**	**96.09**	**83.21**	94.20	**89.62**	**99.76**	**96.63**	**100.00**	**79.26**

Average Performance

	Individual	Weight Averaging	RegMean (Jin et al., 2023)	Task Arithmetic (Ilharco et al., 2023)	Ties-Merging (Yadav et al., 2023)	AdaMerging (Yang et al., 2024b)	EMR-Merging (Huang et al., 2024)	**+TADrop**
Acc	93.98	42.52	68.14	48.89	37.53	60.25	89.59	**91.85**

Table 6: Task-specific and average performance when merging ViT-B/16 models on 30 tasks. **Bold** numbers indicate the best score within each backbone block.

	Chat			**Math**	**Code**	
Method	MMLU	HellaSwag	TruthfulQA	GSM8K	HumanEval	MBPP
Individual	0.4793	0.75	0.31	0.45	0.23	0.2
Ties-Merging (Yadav et al., 2023)	0.4707	0.74	0.27	0.23	**0.29**	**0.29**
+ TADrop(w/o)	0.4768	0.75	0.27	0.38	**0.29**	0.27
+ TADrop	**0.4858**	**0.76**	**0.28**	**0.41**	**0.29**	0.26
EMR-Merging (Huang et al., 2024)	0.4753	0.72	**0.3**	0.47	0.27	0.24
+ TADrop(w/o)	0.4860	0.73	**0.3**	**0.52**	0.27	0.25
+ TADrop	**0.4909**	**0.75**	0.29	**0.52**	**0.29**	**0.27**

Table 7: Results on chat, math, and code benchmarks for Llama-2 models. **Bold** numbers indicate the best score within each backbone block.

Method	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	Avg.
Individual	60.18	94.04	89.22	90.63	91.41	87.20	92.71	79.06	85.56
Weight Averaging	13.96	64.11	69.36	31.84	75.36	42.19	58.70	55.23	51.34
Fisher Merging (Matena and Raffel, 2022)	6.11	53.56	65.21	22.80	58.51	34.80	67.64	55.96	45.57
RegMean (Jin et al., 2023)	36.67	90.60	75.74	62.68	83.55	70.02	82.35	58.48	70.01
Task Arithmetic (Ilharco et al., 2023)	18.78	85.89	**79.90**	**74.03**	83.78	59.08	69.67	62.09	66.65
+ ImPart (Yang et al., 2025a)	21.29	85.67	79.17	72.40	82.24	54.43	69.21	61.37	65.72
+ TALL-masks (Wang et al., 2024)	11.39	79.01	76.72	25.30	81.26	38.60	71.06	53.07	54.55
+ TADrop(w/o)	28.36	85.78	79.17	71.58	**84.11**	59.84	76.26	61.73	68.35
+ TADrop	**30.90**	**86.81**	78.92	68.90	83.89	**62.59**	**81.22**	**64.98**	**69.78**
TIES-Merging (Yadav et al., 2023)	20.48	84.40	**81.13**	58.19	**85.70**	64.65	74.81	42.96	64.04
+ ImPart (Yang et al., 2025a)	9.67	85.21	79.90	69.95	82.09	52.64	59.47	52.71	61.45
+ TALL-masks (Wang et al., 2024)	**31.44**	84.52	78.43	58.21	83.49	57.41	77.78	60.29	66.45
+ TADrop(w/o)	22.67	85.44	78.43	**73.80**	84.67	63.76	73.57	57.76	67.51
+ TADrop	27.43	**86.24**	78.92	72.08	84.48	**66.93**	**79.68**	**61.37**	**69.64**
EMR-Merging (Huang et al., 2024)	39.96	93.35	86.27	82.77	89.72	85.45	89.57	74.37	80.18
+ TADrop(w/o)	**62.04**	93.69	88.48	90.72	90.94	**85.96**	92.28	75.45	84.94
+ TADrop	61.44	**93.92**	**89.22**	**90.76**	**91.12**	85.94	**92.42**	**76.53**	**85.17**

Table 8: per-task performance when merging RoBERTa-base models fine-tuned on eight GLUE text classification tasks. **Bold** numbers indicate the best score within each backbone block, and underlined numbers indicate the second-best score.