Escaping Optimization Stagnation: Taking Steps Beyond Task Arithmetic via Difference Vectors

Jinping Wang, Zhiqiang Gao*, Zhang Dinggen, Zhiwu Xie

College of Science, Mathematics and Technology (CSMT), Wenzhou-Kean University
{1306325, zgao, 1336574, zxie} @wku.edu.cn

Abstract

Current methods for editing pre-trained models face significant challenges, primarily high computational costs and limited scalability. Task arithmetic has recently emerged as a promising solution, using simple arithmetic operations—addition and negation—based on task vectors which are the differences between fine-tuned and pre-trained model weights, to efficiently modify model behavior. However, the full potential of task arithmetic remains underexplored, primarily due to limited mechanisms for overcoming optimization stagnation. To address this challenge, we introduce the notion of difference vector, a generalized form of task vectors derived from the historical movements during optimization. Using difference vectors as directed perturbations, we propose the Difference Vector-based Anisotropic Scaling Iterative algorithm (DV-BASI) to enable a continuous optimization process for task arithmetic methods without relying on any additional modules or components. Notably, by leveraging escapability and directional advantages of difference vectors, the average performance on different tasks of the multi-task model merged by DV-BASI may even outperform models individually fine-tuned. Based on this observation, we extend the application of difference vectors to a feasible fine-tuning method for single-task models. On the practical side, DV-BASI allows expressive searching directions with few learnable parameters and forms a scalable framework. We also integrate DV-BASI with task arithmetic methods and advanced optimization techniques to achieve state-of-the-art performance on both supervised and unsupervised evaluation protocols.

code — https://github.com/smithgun2005/DVBASI

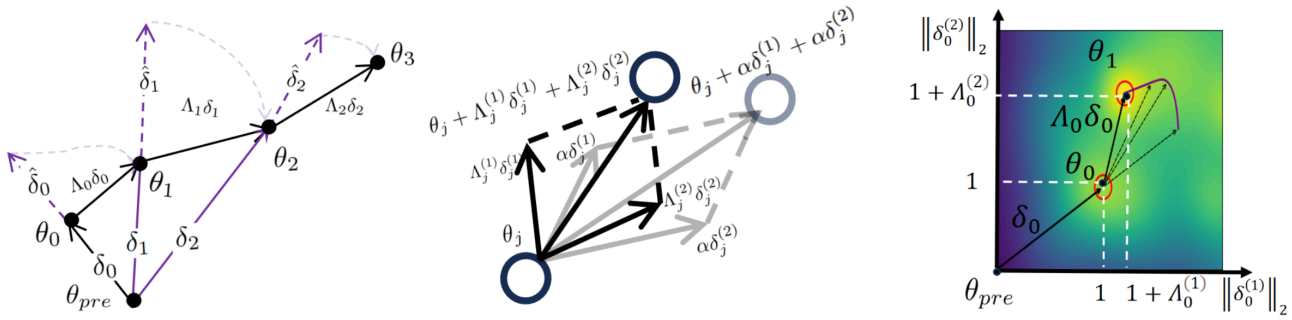
Introduction

Pre-trained models are essential in contemporary machine learning systems due to their efficiency and transferability. Editing models after pre-training is widely recognized as an effective way to enhance model performance on specific downstream tasks (Wortsman et al. 2022; Zhuang et al. 2021; Matena and Raffel 2022), mitigate undesired behaviors (Santurkar et al. 2021; Ribeiro and Lundberg 2022; Murty et al. 2022), align models with human preferences (Askell et al. 2021; Ouyang et al. 2022; Kasirzadeh and Gabriel 2022), or incorporate new information (Cao, Aziz, and Titov 2021; Mitchell et al. 2022a,b). However, traditional editing approaches, which rely on expensive joint fine-tuning across multiple tasks (Vu et al. 2022) and human feedback (Matthews 1975), face limitations in scalability and accessibility. Moreover, optimizing models for downstream tasks often comes at the expense of diminished pre-training performance or zero-shot accuracy (Garipov et al. 2018; Loshchilov and Hutter 2019; Stallkamp et al. 2011a).

Recently, innovative research on task arithmetic has introduced cost-effective and scalable model editing techniques (Ilharco et al. 2023; Yadav et al. 2023; Yang et al. 2024; Ortiz-Jiménez, Favero, and Frossard 2023; Yoshida et al. 2025; Zhang et al. 2024). By leveraging the concept of task vector that is defined as the element-wise difference between the weights of fine-tuned and pre-trained models, task arithmetic can modify various models through simple arithmetic operations on these vectors (Ilharco et al. 2023). Specifically, negating a task vector can eliminate undesirable behaviors on specific tasks (task negation), while adding task vectors from different tasks can lead to the creation of a multi-task model that performs well on multiple tasks simultaneously (task addition). Recent advances on linearized task vectors deepen the theoretical understanding on task arithmetic by addressing the interference among task vectors. Through techniques based on model linearization via the neural tangent kernel approximation (Ortiz-Jiménez, Favero, and Frossard 2023) and τ-Jacobian product regularization (τJp Reg) (Yoshida et al. 2025) during the *model pre-training* stage, the linearized task vectors can be produced with less weight disentanglement error.

Although recent studies have advanced our understanding of task arithmetic, current approaches for designing task vector combination strategies have not yet realized the full potential of task arithmetic. Ideally, a merged multi-task model edited through task arithmetic is expected to achieve performance comparable to that of individually fine-tuned single-task models. However, due to the limited expressive power of combination coefficients learned via coarse grid search (Ilharco et al. 2023; Yadav et al. 2023), this goal remains elusive in practice. Although current finer-grained *Parameter-Efficient Fine-Tuning (PEFT)* task vector combination methods based on block-wise optimization (Zhang

*Corresponding author: zgao@wku.edu.cn.



(a) Overall iterative procedure of DV-BASI (b) Anisotropic scaling of a difference vector (c) Optimization trajectory in the loss landscape

Figure 1. The overall iterative procedure of DV-BASI is illustrated in (a). Starting from pre-trained weights θ_{pre}, the model initially reaches a local optimum θ_0 during the first major optimization step. At each local optimum θ_j, DV-BASI computes difference vectors δ_j (indicated by purple arrows), which provide directional guidance for further optimization ($\hat{\delta}_j$ denotes the directional vector of δ_j). Based on these difference vectors, we apply anisotropic scaling matrices Λ_j to create more flexible exploration directions, aiming to find a potentially better global solution. (b) provides a detailed illustration of the anisotropic scaling process for a difference vector. Assume each difference vector has two parameter blocks $\delta_j = (\delta_j^{(1)}, \delta_j^{(2)})$. Each block is independently scaled by the anisotropic matrix Λ_j (where $\Lambda_j = (\Lambda_j^{(1)}, \Lambda_j^{(2)})$), which offers more expressive searching directions compared to using a scalar scaling coefficient α (Zhang et al. 2024). (c) visualizes the iterative optimization path of DV-BASI in a loss landscape. It demonstrates how difference vectors function as directed perturbations, effectively helping model weights escape from the current local optima (red circles) to continue searching anisotropically (the purple line represents the anisotropic scaling trajectory of DV-BASI, based on gradient descent) for a potentially better solution in the parameter space.

et al. 2024; Yang et al. 2024) are addressing this issue, they are still fundamentally constrained by their single-step nature. Specifically, optimization often stops prematurely when model parameters become trapped in local optima where gradients vanish, thus impeding further exploration. Therefore, analogous to traditional parameter optimization methods, developing a multi-step optimization approach that can efficiently escape local optima and realize continuous optimization to find a better global solution is a crucial task.

To address this challenge, we propose a difference vector-based anisotropic scaling iterative algorithm (DV-BASI) to achieve continuous exploration in the parameter space, as illustrated in Figure 1. We extend the concept of task vectors to a more general difference vector, defined as the element-wise difference between the weights of a model in any arbitrary state during training and those of the pre-trained model. Similar to task vectors as knowledge carriers, the difference vector, as a cumulative result of previous optimizations, contains information of historical movements about the model weights from the training process. Through theoretical and empirical analysis, we demonstrate that difference vectors enable continuous model optimization with the following merits: (i) *Escapability and Directional Advantage*: When model weights are trapped in a local optimum, the updated difference vector at that point acts as a directed perturbation, effectively helping the model weights escape the current critical point and continue searching for a potentially better solution. (ii) *Component-Free Continuity*: Continuous exploration in the parameter space relies solely on the updates of the difference vector, without depending on additional components such as adapters (Houlsby et al. 2019), prompts (Jia et al. 2022), or LoRA (Hu et al. 2022).

We demonstrate that DV-BASI is a scalable multi-step task arithmetic framework. Adhering to the standard evaluation protocols of task arithmetic (Ilharco et al. 2023; Ortiz-Jiménez, Favero, and Frossard 2023; Yoshida et al. 2025) and its extended PEFT paradigm (Zhang et al. 2024; Yang et al. 2024), our framework can seamlessly integrate with existing task arithmetic techniques (for example τJP and aTLAS) and adapt both unsupervised and supervised learning settings, delivering state-of-the-art (SOTA) performance. To further highlight its scalability from an optimization standpoint, we propose a Multi-Objective Optimization (MOO) strategy that treats each task as an independent objective as a case study to extend our learning framework.

The contributions of methods are as follows: (1) We extend the learning paradigm of task arithmetic to a multi-step approach called DV-BASI by employing difference vectors. DV-BASI can effectively enhance task arithmetic performance and realize state-of-the-art (SOTA) performance under both supervised and unsupervised settings. Empirical analyses and theoretical explanations demonstrate that difference vectors possess escapability and directional advantages, enabling conventional methods to escape local optima for further improvement. Additionally, the component-free nature of difference vectors promotes continuous exploration in the parameter space. (2) Leveraging the benefits of our difference vectors, we expand the application scope of task arithmetic to further enhance the performance of already fine-tuned single-task models. (3) The proposed DV-BASI is a novel and scalable framework that can easily integrate with conventional task arithmetic methods. This inte-

gration effectively unleashes their potential, resulting in better performances.

Models and Difference Vectors

Investigations into task arithmetic, as initially explored in (Ilharco et al. 2023), have revealed intriguing attributes of task vectors across diverse models. Following the established setting of aTLAS (Zhang et al. 2024), this study focuses on the CLIP (Radford et al. 2021) model, leveraging its extensive availability and manageable scale to facilitate a deeper analysis. Specifically, we derive task vectors through fine-tuning the image encoder while preserving the text representations. This method ensures that image encoders fine-tuned on distinct datasets produce features within a unified representational space, thanks to a common text encoder. As a result, task vectors from these fine-tuned encoders can be more seamlessly integrated to create a cohesive multi-task model.

Formally, let the CLIP image encoder be represented as $f : \mathcal{X} \times \Theta \rightarrow \mathcal{Z}$, where for an input image $x \in \mathcal{X}$ and parameters $\theta \in \Theta$, $z = f(x;\theta)$ denotes the learned latent representation of the input image. Let the weights of a pre-trained model be θ_{pre}, and the weights of its fine-tuned version be $\theta_{ft}^{(i)}$, with $i \in \mathbb{N}^+$, where i indexes a dataset $\mathcal{D}^{(i)}$. Following Ilharco et al. (Ilharco et al. 2023), we define a task vector as $\tau^{(i)} = \theta_{ft}^{(i)} - \theta_{pre}$.

Generally, task arithmetic methods merge task vectors into a multi-task model by developing efficient weighting schemes to combine them. Given the high-dimensional and complex nature of model parameter spaces, merged models often encounter local optima, where standard training fails to significantly reduce loss or enhance model performance. Garipov et al. (Garipov et al. 2018) have shown that despite the complexity of the loss surfaces of deep neural networks, optimal points are not isolated and can be connected through a simple low-loss path. This finding demonstrates the rationality to continue searching for other possible better solutions as long as along the proper direction.

The difference vector is instrumental in constructing a multi-step approach to assist task arithmetic in escaping local optima for further enhancement. When a merged model becomes temporarily trapped in a local optimum θ^*, the difference vector δ can be defined as the element-wise difference between θ^* and θ_{pre}, that is $\delta = \theta^* - \theta_{pre}$. Since task vectors act as carriers of knowledge (Zhang et al. 2024), difference vectors, as generalized task vectors, also encapsulate the historical model knowledge from the pre-trained to the local optima.

Escaping Optimization Stagnation via DV-BASI

DV-BASI is an extensible multi-step task arithmetic framework that can be used to optimize the models merged in the previous iteration. Starting from the initial optimal point θ_0 the corresponding initial difference vector δ_0 is defined as the element-wise difference between θ_0 and θ_{pre}. *Notably, the initial optimal point θ_0 represents the parameters of a model that have been merged by pre-defined methods, such as aTLAS and τJP reg as shown in our Experiments.* Here, the first iteration is formulated as:

$$\delta_0 = \theta_0 - \theta_{\text{pre}}, \quad \theta_1 = \theta_0 + \Lambda_0 \delta_0, \tag{1}$$

where θ_1 represents the updated model parameters, and Λ_0 is a learnable anisotropic scaling matrix, which will be detailed in Equations 3 and 4.

During each iteration process, if the model's weights get stuck in a local optimal solution that is difficult to break through (manifested as no improvement in accuracy over several epochs), the best-performing parameters will be selected as the starting point for the subsequent iteration. Here, the difference vector is updated to serve as a directed perturbation, propelling the model parameters away from this local optimum to facilitate ongoing exploration. Generally, in the $(j+1)$-th iteration, the difference vector δ_j is updated based on θ_j, and the update for the next optimum θ_{j+1} is given by:

$$\delta_j = \theta_j - \theta_{\text{pre}}, \quad \theta_{j+1} = \theta_j + \Lambda_j \delta_j. \tag{2}$$

To effectively explore the next optimum in each iteration, an anisotropic scaling mechanism is applied to the difference vectors, enabling flexible and controllable exploration within the parameter space. Typically, parameters across different layers of neural networks have distinct roles and functionalities. Inspired by Zhang et al. (Zhang et al. 2024), instead of using a scalar α to scale the difference vector, we decompose the difference vector into parameter blocks $\delta = (\delta^{(1)}, \delta^{(2)}, ..., \delta^{(n)})$ and assign each block an independent learnable scaling coefficient for anisotropic exploration in the parameter space. Consequently, we introduce a block diagonal scaling matrix Λ:

$$\Lambda = \begin{bmatrix} \lambda^{(1)} I^{(1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda^{(n)} I^{(n)} \end{bmatrix}, \tag{3}$$

where $\lambda^{(i)}$ denotes the coefficient for each block, and $I^{(i)}$ represents the identity matrix. This results in anisotropic scaling of the difference vector, expressed as:

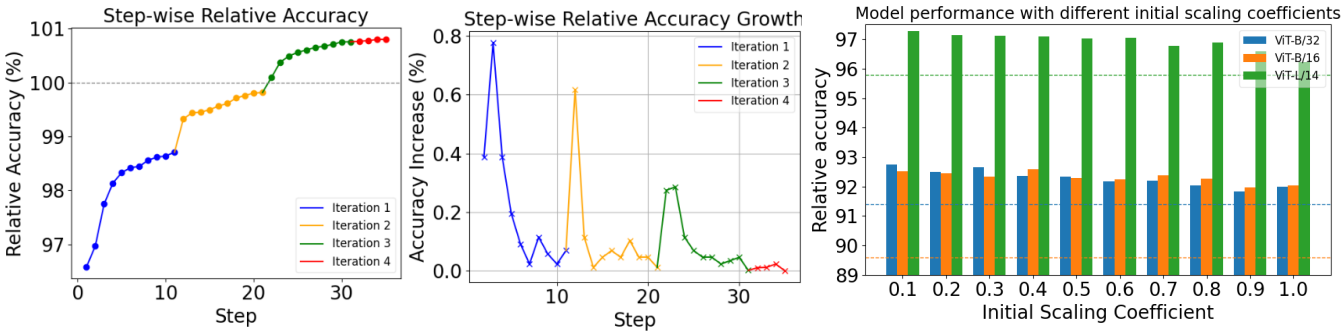
$$\Lambda\delta = (\Lambda^{(1)}\delta^{(1)}, \Lambda^{(2)}\delta^{(2)}, ..., \Lambda^{(n)}\delta^{(n)}). \tag{4}$$

Algorithm 1: **DV-BASI**

Input: Pre-trained weights θ_{pre}, initial weights θ_0 (obtained by pre-defined model merging methods), learning rate η, number of iterations M

Output: Final weights θ_M

for $m = 1$ to M **do**
  $\delta_{m-1} \leftarrow \theta_{m-1} - \theta_{\text{pre}}$ // Difference vector
  Initialize $\Lambda_{m-1}^{(0)}$
  for $t = 0, 1, \ldots$ until early stopping **do**
    $\Lambda_{m-1}^{(t+1)} \leftarrow \Lambda_{m-1}^{(t)} - \eta \nabla_{\Lambda_{m-1}^{(t)}} \mathcal{L}(\theta_{m-1}^{(t)})$ // Solve Eq.(5)
  end for
  Let Λ_{m-1}^{*} be the converged scaling matrix
  $\theta_m \leftarrow \theta_{m-1} + \Lambda_{m-1}^{*} \delta_{m-1}$
end for



(a) (b) (c)

Figure 2. Figure (a) and (b) show the stepwise relative accuracy of supervised model merging (using ViT-B/32 as pre-trained backbone) and its growth within 4 DV-BASI iterations. Figure (c) compares the unsupervised model merging performance of 10 different initial scaling coefficients (0.1 to 1.0) among 3 pre-trained backbones (ViT-B/32, ViT-B/16, and ViT-L/14).

Method	ViT-B/32	ViT-B/16	ViT-L/14
Initial weights (θ_0)	82.9	82.8	90.2
Random perturbation ($\theta_{j+1} = \theta_j + \Lambda_j \|\delta_j\| R_j$)	5.6	6.3	4.2
Isotropic (scalar) scaling ($\theta_{j+1} = \theta_j + \alpha_j \delta_j$)	83.3	83.6	90.3
Anisotropic scaling ($\theta_{j+1} = \theta_j + \Lambda_j \delta_j$)	**83.9**	**85.3**	**90.8**

Table 1. Unsupervised model merging performance comparison between different types of perturbations and scaling strategies. R_j denote the unit random vector.

The anisotropic scaling matrix Λ is the sole learnable parameter within each iteration. Assuming a supervised learning context, the optimization problem for the $(j+1)$-th iteration is formulated as:

$$\arg\min_{\Lambda_j} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\mathcal{L}\big(f(x;\,\theta_j + \Lambda_j\delta_j),\,y\big)\Big], \qquad (5)$$

where $\mathcal{L}$ denotes the cross entropy loss function for the target task, and $\mathcal{D} = \{\mathcal{D}^{(i)}\}_{i=1}^{n}$ represents n target datasets from which (x, y) is drawn. Notably, when adapting DV-BASI to an unsupervised learning scenario, the detailed unsupervised loss function can be found in the Appendix. The iterative learning process is introduced in Algorithm 1.

The DV-BASI is a scalable framework, that allows an advanced optimization paradigm, can be flexibly integrated into other task arithmetic approaches. We introduce a Multi-Objective Optimization (MOO) approach, treating each task as a distinct objective and facilitating balanced optimization across multiple tasks. By employing the concept of Pareto optimality, we utilize the Multiple Gradient Descent Algorithm (MGDA) (Désidéri 2012) to ascertain the most balanced optimization direction, effectively minimizing losses across tasks as parallel objectives. The specifics of our MOO algorithm are elaborated in the Appendix.

Empirical Analysis

Difference vector is the cumulative result of all successful optimization steps taken so far, containing information about directions that have successfully reduced the loss in the previous training process. By applying difference vectors as perturbations, we can deliberately push the weights along the direction known to be effective, thereby escaping the current equilibrium in a guided manner. In this section, we will further discuss the escapability of the difference vectors and directional advantages compare to random perturbations.

Escapability: Difference vectors, functioning as directed perturbations, can effectively push the model weights away to escape the optimization stagnation points. To empirically verify its escapability, we illustrate the stepwise relative accuracy (accuracy of merged model divided by that of the fine-tuned models) and its growth interval, in Figures 2 (a) and (b), where all results are produced under a supervised learning setting. We experiment with 4 iterations, within the same iteration, the accuracies are marked with the same color, and the value of each point represents the result obtained after each training epoch. In (a), after applying DV-BASI, the relative accuracy iteratively increases and finally outperforms 100%. This observation means that the performances of the merged model exceed those of fine-tuned models. In (b), we can observe that during the early stages of each iteration, when the difference vectors are updated, the performances improve significantly, but as training progresses, the improvement gradually stagnates. When we update the difference vector at the beginning of each iteration, the previously stagnated optimization can resume. These results explicitly illustrate that our method effectively helps the model to escape from local stagnation and results in continuous improvement.

Unlike random perturbations, difference vectors as knowledge carriers contain knowledge learned from previous training steps.

T.V.	Methods	ViT-B/32		ViT-B/16		ViT-L/14	
		Target (↓)	Control (↑)	Target (↓)	Control (↑)	Target (↓)	Control (↑)
n/a	Pre-trained	48.1	63.4	55.5	68.3	64.9	75.5
Unsupervised							
Std.	Task arithmetic	24.0	60.9	21.3	65.4	19.0	72.9
	Ties-Merging	21.8	**61.7**	24.3	**67.0**	26.6	**74.4**
	aTLAS	23.3	60.7	21.0	65.0	17.8	73.2
	aTLAS + DV-BASI	**20.3**$^{-1.5}$	60.2	**20.2**$^{-0.8}$	65.0	**15.2**$^{-2.6}$	72.5
Lin.	Task arithmetic	10.9	60.8	11.3	64.8	7.9	72.5
	τJP reg	6.7	60.8	4.7	**66.0**	3.7	73.0
	τJP reg + DV-BASI	**5.7**$^{-1.0}$	**60.8**	**4.4**$^{-0.3}$	65.0	**3.6**$^{-0.1}$	**73.2**
Supervised							
Std.	aTLAS*	19.4	**61.2**	18.1	**65.8**	17.8	**73.3**
	aTLAS* + DV-BASI*	**10.7**$^{-8.7}$	60.6	**14.5**$^{-3.6}$	64.9	**12.6**$^{-5.2}$	72.5
Lin.	aTLAS*	11.1	61.0	10.2	**65.6**	12.6	73.1
	aTLAS*+ DV-BASI*	9.5$^{-1.6}$	**61.3**	8.4$^{-1.8}$	65.0	11.2$^{-1.4}$	73.2
	τJP reg + DV-BASI*	**4.1**$^{-7.0}$	61.0	**3.6**$^{-6.6}$	65.4	**2.1**$^{-10.5}$	**73.6**

Table 2. Performances of task negation averaged across eight datasets. All our results maintain at least 95% of the pre-trained accuracy on the control dataset. Results marked with an asterisk * indicate the supervised setting.

Denote k as the step index, K as the total number of training steps to reach the local optimal point θ^*, and $\theta^{(k)}$ represents the model weights at k_{th} step, we have:

$$\delta = \theta^* - \theta_{\text{pre}} = \sum_{k=1}^{K} \Delta\theta^{(k)}, \quad \Delta\theta^{(k)} = \theta^{(k)} - \theta^{(k-1)}. \tag{6}$$

To further investigate the effect of the initial scaling coefficient α for each parameter block on the performance, we evaluate the merged model's under an unsupervised setting with different initial α values (from 0.1 to 1.0). As shown in Figure 2 (c), we observe that, across the range of initial scaling coefficients, DV-BASI consistently improves model performance compared to the initial pre-trained weights. This suggests that as long as the perturbation is applied in a proper direction, DV-BASI is insensitive to α and it is possible to converge to a better solution. Intuitively, the observations may mainly be attributed to the directional advantage of the difference vector, which leads to an exploration of the following directional properties of difference vectors.

Directional Advantage: Compared to random perturbations, using difference vectors as perturbations has a directional advantage. They always point in the direction where model weights improved in previous optimization steps. To demonstrate the advantage of perturbing along the direction of difference vectors, we compare the model performance with that of a model using random perturbations (with the same magnitude as our difference vectors) under an unsupervised model merging setting. Meanwhile, we also compare model performances under anisotropic and isotropic scaling strategies, respectively. As observed in Table 1, unlike difference vectors, random perturbations lead to a catastrophic drop in performance, indicating the necessity of perturbing along the direction of difference vectors. *A more theoretical perspective on why this phenomenon exists (Random perturbations can degrade model performance, whereas difference vectors contribute to continued optimization) is illustrated in the Appendix.* Meanwhile, the results show that compared to isotropic scaling, scaling the difference vector anisotropically is more likely to realize a better model performance due to its flexibility in searching directions.

Our difference vector functions similarly to the momentum term in traditional gradient-based optimization, where the update direction from the historical optimization steps serves as an inertial force to help the model escape from the current local minimum (where gradients vanish). However, like momentum, the direction provided by the difference vector does not guarantee convergence to a better global optimum. In fact, in the context of task arithmetic, referencing a global historical update direction is often significantly more effective than using a random one 1 with significantly higher probability of reaching a better solution.

Computational Efficiency Due to the iterative nature of DV-BASI, our method introduces some additional training time and storage overhead. While it is difficult to fairly compare all methods because of reproducibility issues, we emphasize that DV-BASI achieves comparable computational efficiency to current advanced approaches. In terms of training time, compared with fine-tuning-based methods such as Adamerging (Yang et al. 2024) and aTLAS (Zhang et al. 2024) that both require learning sophisticated merging coefficients during training, our method directly takes the merged model as input and only introduces training time for refining it. Therefore, DV-BASI has comparable runtime to these methods. For pre-training-based methods (e.g., τJp Reg (Yoshida et al. 2025), Linearized Task Vector (Ortiz-Jiménez, Favero, and Frossard 2023)) that require substantial time to train each task-specific vector, DV-BASI is more efficient in terms of total time consumed. Regarding storage, DV-BASI only needs to maintain one merged model and perform 3–5 refinement iterations. This results in storing only 4–6 models in total, which is significantly more resource-efficient than existing methods. In summary, although DV-

T.V.	Methods	ViT-B/32		ViT-B/16		ViT-L/14	
		Abs. (↑)	Rel. (↑)	Abs. (↑)	Rel. (↑)	Abs. (↑)	Rel. (↑)
n/a	Pre-trained	48.1	–	55.5	–	64.9	–
unsupervised							
Std.	Task arithmetic	70.1	77.2	73.6	79.9	82.9	87.9
	Ties-Merging	74.2	84.8	78.6	87.6	85.0	91.9
	AdaMerging	80.1	88.5	82.9	89.7	90.8	96.4
	aTLAS	82.9	91.4	82.8	89.6	90.2	95.8
	aTLAS + DV-BASI	**83.9**$^{+1.0}$	**92.8**$^{+1.4}$	**85.3**$^{+2.5}$	**92.5**$^{+2.9}$	**90.8**$^{+0.6}$	**96.4**$^{+0.6}$
Lin.	Task arithmetic	74.7	85.2	77.5	86.2	84.8	91.9
	τJP reg	84.5	97.6	87.6	98.1	90.8	99.0
	τJP reg + DV-BASI	**86.7**$^{+2.2}$	**99.8**$^{+2.2}$	**88.0**$^{+0.4}$	**98.6**$^{+0.5}$	**91.8**$^{+1.0}$	**100.1**$^{+1.1}$
Supervised							
Std.	aTLAS*	84.1	92.8	82.9	89.7	91.4	97.1
	aTLAS* + DV-BASI*	85.9$^{+1.8}$	94.8$^{+2.0}$	87.2$^{+4.3}$	94.6$^{+4.9}$	91.6$^{+0.2}$	97.4$^{+0.5}$
	aTLAS* + DV-BASI* + MOO*	**86.2**$^{+2.1}$	**95.1**$^{+2.3}$	**87.7**$^{+4.8}$	**95.1**$^{+5.4}$	**91.8**$^{+0.4}$	**97.6**$^{+0.5}$
Lin.	aTLAS*	83.4	95.4	85.4	95.1	88.7	96.1
	aTLAS* + DV-BASI*	86.6$^{+3.2}$	99.1$^{+3.7}$	87.5$^{+2.1}$	97.4$^{+2.3}$	90.0$^{+1.3}$	97.5$^{+1.4}$
	τJP reg + DV-BASI*	**87.5**$^{+4.1}$	**100.8**$^{+5.4}$	**89.1**$^{+3.7}$	**99.8**$^{+3.7}$	**92.3**$^{+3.6}$	**100.6**$^{+4.5}$

Table 3. Performances of task addition averaged across eight datasets. We report absolute accuracy (Abs.) and relative accuracy (Rel.) with respect to the average accuracy of model fine-tuned on single tasks. Results marked with an asterisk * indicate the supervised setting.

BASI introduces moderate additional computation, it still maintains a comparable and controllable level of computational cost.

Experiments

This section shows the effectiveness of improving task arithmetic performance by applying our DV-BASI algorithm under both supervised and unsupervised settings. Our experiments are conducted under both supervised and unsupervised conditions, **further experimental results and details are included in the Appendix**. For a supervised condition, we use normal cross-entropy as our loss function. For unsupervised situations, following Yang et al. (Yang et al. 2024), we use entropy minimization as an optimization surrogate objective function to find the best group of coefficients each iteration (details included in Appendix).

Methods	ViT-B/32	ViT-B/16	ViT-L/14
Zeroshot	60.40	65.05	72.88
aTLAS	65.16	69.60	75.30
DV-BASI	**66.76**	**70.84**	**76.01**

Table 4. Performance of test-time adaptation

Datasets Following the typical evaluation procedure (Ilharco et al. 2023; Yoshida et al. 2025; Ortiz-Jiménez, Favero, and Frossard 2023; Yang et al. 2024; Zhang et al. 2024), we focus on the computer vision task and apply our DV-BASI to 8 image classification tasks: Cars (Krause et al. 2013), DTD (Cimpoi et al. 2014), EuroSAT (Helber et al. 2018), GTSRB (Stallkamp et al. 2011b), MNIST (Deng 2012), RESISC45 (Cheng, Han, and Lu 2017), SUN397 (Xiao et al. 2016), and SVHN (Netzer et al. 2011). For task negation, we add ImageNet as a control dataset.

Compared Methods We conduct our experiment under supervised and unsupervised conditions. For supervised conditions, we choose aTLAS as our baseline method. For unsupervised task-arithmetic based approaches, we choose a training-free method (original task arithmetic with standard task vectors (Ilharco et al. 2023) and linearized task vectors (Ortiz-Jiménez, Favero, and Frossard 2023) and Ties-Merging (Yadav et al. 2023)) as our baseline method. AdaMerging (Yang et al. 2024) and aTLAS (Zhang et al. 2024) (We use the entropy minimization mentioned before to modify an unsupervised version of aTLAS) are selected as our Train-based baseline method.

Implementation The experiments are performed on NVIDIA GeForce RTX 4090 GPUs. For the initial model state θ_0, we use our reproduced result on different baseline methods. Both supervised and unsupervised conditions are performed by taking ViT-B/32, ViT-B/16, and ViT-L/14 architectures in CLIP (Radford et al. 2021) as backbones. All models are optimized by applying the AdamW (Loshchilov and Hutter 2017) optimizer with a learning rate of 0.01. To identify local minima, we adopt a commonly used criterion based on the model's performance on the validation/test set. Within 60 epochs, we use early stopping with a patience of 5 epochs to judge the best models (local optima points in parameter space), and then use such a best model to update the difference vector to search continuously.

Task Negation Task negation aims to reduce the model's performance on a target task while maintaining performance on a control task. Following the standard evaluation procedure, the model is expected to *forget as much as possible on the target task under the constraint of maintaining at least 95% performance on the control task*.

Denote the validation set for the target task by $\mathcal{D}_t$ and the control task by $\mathcal{D}_c$. We apply a simultaneous gradient

Datasets	Cars	DTD	RESISC45	SUN	Food101	ImageNet	Caltech256	PascalVOC	Country221	UCF101
Finetune	78.26	78.94	95.94	75.40	88.58	76.41	92.60	88.42	21.99	85.01
DV-BASI	80.30	78.94	96.05	75.65	89.70	78.40	92.73	89.48	23.70	85.20

CIFAR10	SVHN	CIFAR100	FGVCAircraft	Flowers	OxfordPet	CUB200	UCF101	Caltech101	EuroSAT	**AVG**
98.05	97.38	89.09	40.70	90.08	92.15	73.56	85.01	94.41	98.89	82.04
98.35	97.42	89.09	42.90	91.95	92.21	73.56	85.14	94.47	99.70	82.74

Table 5. Results of applying DV-BASI after fine-tuning on different 20 tasks. CLIP with the ViT-B/32 backbone is used.

descent on the control task and gradient ascent on the target task. Thus, the optimization problem in $(j+1)$-th iteration can be described as:

$$\begin{aligned}\arg\min_{\Lambda_j} \mathbb{E}_{(x,y)\in\mathcal{D}_t}\left[-\mathcal{L}\left(f(x;\theta_j+\Lambda_j\delta_j),y\right)\right] \\ +\mathbb{E}_{(x,y)\in\mathcal{D}_c}\left[\mathcal{L}\left(f(x;\theta_j+\Lambda_j\delta_j),y\right)\right].\end{aligned} \tag{7}$$

The findings presented in Table 2 demonstrate that DV-BASI significantly mitigates undesired biases (i.e., reducing performances of the target tasks) while sustaining over 95% accuracy on control tasks. Under supervised and unsupervised conditions, for both linearized task vectors and standard task vectors, DV-BASI can effectively improve task negation and achieve SOTA performances.

Task Addition Task addition aims to create a multi-task model on several target datasets by adding task vectors from those target datasets. This operation allows us to transfer or reuse the knowledge from models individually fine-tuned on specific tasks. Moreover, we embed the MOO algorithm we proposed and the regularization of τ JP into our DV-BASI in the cases of standard task vector and linearized task vector.

As illustrated in Table 3, DV-BASI enhances the performance of multi-task models merged through task addition. Specifically, DV-BASI achieves SOTA performance in both supervised and unsupervised task addition settings with both linearized and standard task vectors. Notably, when DV-BASI is applied to linearized task vectors, the relative accuracy may even exceed 100 percent. This highlights that DV-BASI fully utilizes the potential of task vectors and may even surpass the performances of fine-tuned models.

Test-Time Adaptation Test-time adaptation (TTA) (Liang, Hu, and Feng 2020; Sun et al. 2020; Wang et al. 2020) operates under the premise of absent labeled data for the target task, focusing on bolstering model robustness against domain shifts and out-of-distribution scenarios. Specifically, let $T = \{\tau^{(i)}\}_{i=1}^{n}$ represent the collection of task vectors for all accessible target sets, with $\mathcal{D}^{(i)}$ denoting the corresponding dataset for task vector $\tau^{(i)}$. For each target dataset $\mathcal{D}^{(i)}$, the subset $T \setminus \{\tau^{(i)}\}$ is utilized to learn composition and mitigate knowledge leakage. We adhere to the experimental setup of (Zhang et al. 2024), conducting offline adaptation on 22 image classification datasets (further details in the Appendix) with ViT-B/32, ViT-B/16, and ViT-L/14 architectures from CLIP (Radford et al. 2021) serve as our backbone models. As shown in Table 4, DV-BASI delivers superior performance across all models, achieving 66.76, 70.84, and 76.01, respectively, outperforming aTLAS by 1.60, 1.24, and 0.71 points. These findings underscore DV-BASI's proficiency in knowledge transfer, facilitating more effective adaptation in the absence of labeled data.

Enhancing Single-Task Performance Beyond Fine-Tuning Inspired by the observation that the performance of the multi-task model merged by DV-BASI may outperform fine-tuned models, we wonder if DV-BASI is applicable to further improve model performance after fine-tuning on different datasets. Thus, we treat the fine-tuned weights of each dataset as the initial weights θ_0 and apply DV-BASI to explore the possibility of further improvement. In our experiment, we chose 20 datasets from 22 datasets (due to the fine-tune accuracies on both MNIST and GTSRB already exceeding 99 percent) in the previous TTA experiment, and ViT-B-32 in CLIP (Radford et al. 2021) as the backbone. All fine-tuned weights of 20 datasets are obtained from aTLAS (Zhang et al. 2024). As shown in Table 5, the average accuracy among 20 datasets increases after applying DV-BASI. In detail, for 17 out of 20 tasks, model performances can be further improved. This demonstrates the potential of leveraging the difference vector for single-task tuning. Therefore, we believe that editing atuningune models with the difference vectors is very promising and has great potential for future development.

Conclusion

In this paper, we introduced DV-BASI, a novel multi-step optimization framework that extends the paradigm of task arithmetic by leveraging difference vectors as directed perturbations. Unlike traditional single-step arithmetic or parameter-efficient fine-tuning methods, DV-BASI offers a component-free and scalable way to continuously escape optimization stagnation through anisotropic scaling of difference vectors. Our theoretical and empirical analysis highlighted several properties of DV-BASI. We conducted experiments across task arithmetic, test-time adaptation, and single-task tuning to demonstrate the validity of our approach with supervised and unsupervised objectives.

Acknowledgements

The work was partially supported by the following: WKU 2025 Summer Student Partnering with Faculty Research Program under No. SSPF2025022, WKU Internal (Faculty/Staff) Start-up Research Grant under No. ISRG2024009, WKU 2025 International Collaborative Research Program under No. ICRPSP2025001.

References

Askell, A.; Bai, Y.; Chen, A.; Drain, D.; Ganguli, D.; Henighan, T.; Jones, A.; Joseph, N.; Mann, B.; DasSarma, N.; Elhage, N.; Hatfield-Dodds, Z.; Hernandez, D.; Kernion, J.; Ndousse, K.; Olsson, C.; Amodei, D.; Brown, T. B.; Clark, J.; McCandlish, S.; Olah, C.; and Kaplan, J. 2021. A General Language Assistant as a Laboratory for Alignment. *CoRR*, abs/2112.00861.

Cao, N. D.; Aziz, W.; and Titov, I. 2021. Editing Factual Knowledge in Language Models. In Moens, M.; Huang, X.; Specia, L.; and Yih, S. W., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, 6491–6506. Association for Computational Linguistics.

Cheng, G.; Han, J.; and Lu, X. 2017. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE*, 105(10): 1865–1883.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing Textures in the Wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 3606–3613. IEEE Computer Society.

Deng, L. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Process. Mag.*, 29(6): 141–142.

Désidéri, J.-A. 2012. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6): 313–318.

Garipov, T.; Izmailov, P.; Podoprikhin, D.; Vetrov, D. P.; and Wilson, A. G. 2018. Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 8803–8812.

Helber, P.; Bischke, B.; Dengel, A.; and Borth, D. 2018. Introducing Eurosat: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. In *2018 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2018, Valencia, Spain, July 22-27, 2018*, 204–207. IEEE.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; de Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-Efficient Transfer Learning for NLP. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 2790–2799. PMLR.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Ilharco, G.; Ribeiro, M. T.; Wortsman, M.; Schmidt, L.; Hajishirzi, H.; and Farhadi, A. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *European conference on computer vision*, 709–727. Springer.

Kasirzadeh, A.; and Gabriel, I. 2022. In conversation with Artificial Intelligence: aligning language models with human values. *CoRR*, abs/2209.00731.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3D Object Representations for Fine-Grained Categorization. In *2013 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2013, Sydney, Australia, December 1-8, 2013*, 554–561. IEEE Computer Society.

Liang, J.; Hu, D.; and Feng, J. 2020. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, 6028–6039. PMLR.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Matena, M.; and Raffel, C. 2022. Merging Models with Fisher-Weighted Averaging. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *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*.

Matthews, B. W. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2): 442–451.

Mitchell, E.; Lin, C.; Bosselut, A.; Finn, C.; and Manning, C. D. 2022a. Fast Model Editing at Scale. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Mitchell, E.; Lin, C.; Bosselut, A.; Manning, C. D.; and Finn, C. 2022b. Memory-Based Model Editing at Scale. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 15817–15831. PMLR.

Murty, S.; Manning, C. D.; Lundberg, S. M.; and Ribeiro, M. T. 2022. Fixing Model Bugs with Natural Language Patches. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 11600–11613. Association for Computational Linguistics.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 4.

Ortiz-Jiménez, G.; Favero, A.; and Frossard, P. 2023. Task Arithmetic in the Tangent Space: Improved Editing of Pre-Trained Models. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *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*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *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*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Ribeiro, M. T.; and Lundberg, S. M. 2022. Adaptive Testing and Debugging of NLP Models. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 3253–3267. Association for Computational Linguistics.

Santurkar, S.; Tsipras, D.; Elango, M.; Bau, D.; Torralba, A.; and Madry, A. 2021. Editing a classifier by rewriting its prediction rules. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 23359–23373.

Stallkamp, J.; Schlipsing, M.; Salmen, J.; and Igel, C. 2011a. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks, IJCNN 2011, San Jose, California, USA, July 31 - August 5, 2011*, 1453–1460. IEEE.

Stallkamp, J.; Schlipsing, M.; Salmen, J.; and Igel, C. 2011b. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks, IJCNN 2011, San Jose, California, USA, July 31 - August 5, 2011*, 1453–1460. IEEE.

Sun, Y.; Wang, X.; Liu, Z.; Miller, J.; Efros, A.; and Hardt, M. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, 9229–9248. PMLR.

Vu, T.; Lester, B.; Constant, N.; Al-Rfou', R.; and Cer, D. 2022. SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 5039–5059. Association for Computational Linguistics.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; and Darrell, T. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Wortsman, M.; Ilharco, G.; Kim, J. W.; Li, M.; Kornblith, S.; Roelofs, R.; Lopes, R. G.; Hajishirzi, H.; Farhadi, A.; Namkoong, H.; and Schmidt, L. 2022. Robust fine-tuning of zero-shot models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, 7949–7961. IEEE.

Xiao, J.; Ehinger, K. A.; Hays, J.; Torralba, A.; and Oliva, A. 2016. SUN Database: Exploring a Large Collection of Scene Categories. *Int. J. Comput. Vis.*, 119(1): 3–22.

Yadav, P.; Tam, D.; Choshen, L.; Raffel, C. A.; and Bansal, M. 2023. TIES-Merging: Resolving Interference When Merging Models. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *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*.

Yang, E.; Wang, Z.; Shen, L.; Liu, S.; Guo, G.; Wang, X.; and Tao, D. 2024. AdaMerging: Adaptive Model Merging for Multi-Task Learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yoshida, K.; Naraki, Y.; Horie, T.; Yamaki, R.; Shimizu, R.; Saito, Y.; McAuley, J.; and Naganuma, H. 2025. Mastering Task Arithmetic: τ-Jp as a Key Indicator for Weight Disentanglement. In *The Thirteenth International Conference on Learning Representations*.

Zhang, F. Z.; Albert, P.; Opazo, C. R.; van den Hengel, A.; and Abbasnejad, E. 2024. Knowledge Composition using Task Vectors with Learned Anisotropic Scaling. In Globersons, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2021. A Comprehensive Survey on Transfer Learning. *Proc. IEEE*, 109(1): 43–76.