ICM-Fusion: In-Context Meta-Optimized LoRA Fusion for Multi-Task Adaptation

Yihua Shao[1,2], Xiaofeng Lin[3], Xinwei Long[4], Siyu Chen[2], Minxi Yan[5], Yang Liu[6], Ziyang Yan[7], Ao Ma[8*], Hao Tang[9], Jingcai Guo[1†]

[1] The Hong Kong Polytechnic University
[2] Institute of Automation, Chinese Academy of Sciences
[3] Guangdong University of Technology
[4] Tsinghua University
[5] The Chinese University of Hong Kong
[6] Beijing Institute for General Artificial Intelligence
[7] University of Trento
[8] JD.com
[9] Peking University
yihuajerry@gmail.com, jc-jingcai.guo@polyu.edu.hk

Abstract

Enabling multi-task adaptation in pre-trained Low-Rank Adaptation (LoRA) models is crucial for enhancing their generalization capabilities. Most existing pre-trained LoRA fusion methods decompose weight matrices, sharing similar parameters while merging divergent ones. However, this paradigm inevitably induces inter-weight conflicts and leads to catastrophic domain forgetting. While incremental learning enables adaptation to multiple tasks, it struggles to achieve generalization in few-shot scenarios. Consequently, when the weight data follows a long-tailed distribution, it can lead to forgetting in the fused weights. To address this issue, we propose **In-Context Meta LoRA Fusion (ICM-Fusion)**, a novel framework that synergizes meta-learning with in-context adaptation. The key innovation lies in our task vector arithmetic, which dynamically balances conflicting optimization directions across domains through learned manifold projections. ICM-Fusion obtains the optimal task vector orientation for the fused model in the latent space by adjusting the orientation of the task vectors. Subsequently, the fused LoRA is reconstructed by a self-designed **Fusion VAE (F-VAE)** to realize multi-task LoRA generation. We have conducted extensive experiments on visual and linguistic tasks, and the experimental results demonstrate that ICM-Fusion can be adapted to a wide range of architectural models and applied to various tasks. Compared to the current pre-trained LoRA fusion method, ICM-Fusion fused LoRA can significantly reduce the multi-tasking loss and can even achieve task enhancement in few-shot scenarios.

Introduction

Multiple LoRA fusion enables efficient multi-task adaptation for large language models after training (Shao et al. 2024). Existing methods (Zhang et al. 2023; Wang et al. 2024) predominantly rely on SVD (Koren 2008) low-rank decomposition for information interpolation. However, these approaches cause damage to LoRA's intrinsic structure during interpolation. Shared-weight methods can mitigate this degradation (Huang et al. 2023), but incur knowledge forgetting when handling significantly divergent weights (Zhang and Li 2024). While VAE (Kingma, Welling et al. 2013) sampling offers information protection, and methods like ICM-LoRA (Shao et al. 2025) have attempted model fusion in latent states, they still incur substantial information loss at deeper levels. Therefore, we introduce meta-learning (Hospedales et al. 2021) into the inter-weight space, enhancing individual LoRA weights' capacity for multi-task adaptation.

Meanwhile, incremental learning (Gepperth and Hammer 2016; Masana et al. 2022) and MoE (Mixture of Experts) (Zoph et al. 2022) reduce knowledge forgetting in multi-task LoRA training. Yet their generalizability in few-

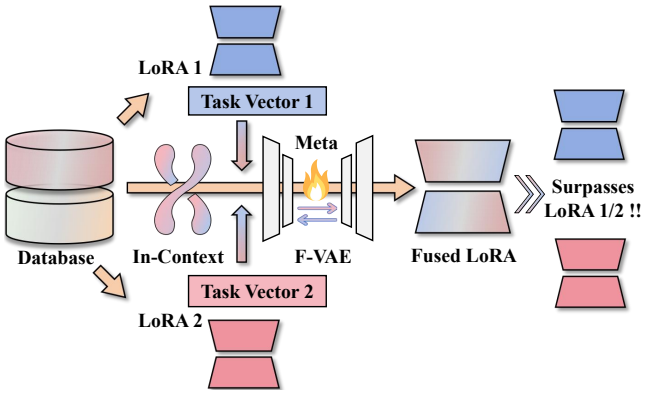

Figure 1: ICM-Fusion enables multi-task generation of LoRA parameters via dynamic task vector arithmetic and context modeling, enhancing model generalization across diverse domains and few-shot scenarios.

*Project leader.
†Correspondence to Jingcai Guo

shot scenarios remains limited (Jiang et al. 2024). For instance, MoE-LoRA (Wu, Huang, and Wei 2024) significantly mitigates forgetting but fails to address data scarcity in few-shot tasks. Furthermore, incremental learning and MoE approaches only achieve adaptation within their predefined task scope, struggling to enhance model capability on out-of-domain generalization tasks (Wu et al. 2024).

To address this, we propose the **In-Context Meta LoRA Fusion** method, which introduces a Fusion Variational Autoencoder (VAE) framework combined with contextual meta-learning to achieve seamless fusion of LoRA parameters. Task vectors, extracted from the hidden states of pretrained models, guide the VAE in encoding parameters into a unified latent space, aligning distributions to reduce conflicts and forgetting. Meanwhile, contextual meta-learning optimizes the latent representations, enhancing the model's generalization ability and enabling rapid adaptation to new tasks with low computational overhead.

Our main contributions can be summarized as follows:

- We propose a novel framework that efficiently merges multiple LoRA models without relying on original datasets, significantly improving parameter storage and inference efficiency.
- Through semantic guidance from task vectors, we achieve alignment and fusion of task-specific parameters, alleviating model conflicts and online forgetting.
- Our method demonstrates superior performance over baseline approaches in multi-task scenarios while maintaining low computational complexity and resource demands.

Related Work

Model Fusion

Model fusion has emerged as a critical area of research in machine learning, enabling the efficient combination of multiple models into a single, more powerful and efficient model (Li et al. 2023; Tang et al. 2024). Early works, such as Model Soups (Wortsman et al. 2022a), demonstrated that averaging weights of independently trained models can improve out-of-distribution (OOD) performance (Jang, Yun, and Han 2024; Wortsman et al. 2022b). Other foundational techniques include Singular Value Decomposition (SVD) (Koren 2008) for model compression and fusion, with extensions like SVD in Latent State that optimize merging in reduced-dimensional spaces. Regularization-based methods such as RegMean (Jin et al. 2022) align models through parameter-space averaging under feature constraints. Recent studies have extended these paradigms to multi-task learning (Caruana 1997; Vandenhende et al. 2021; Zhang et al. 2022). LoRA merging involves combining multiple LoRA (Hu et al. 2022) adapters to create a more versatile model. For example, Meta LoRA (Li et al. 2025b) extends the idea of low-rank adaptation by incorporating meta-learning methods, enabling models to quickly adapt to new tasks with minimal data and computation. TIES-MERGING (Yadav et al. 2023) efficiently merges models by addressing interference in parameter values, involving pruning of task vectors, selection of parameter symbols, and disjoint fusion to improve multitask performance. This was further enhanced by DARE-TIES (Yu et al. 2024), which applies sparsification and rescaling to task vectors before TIES merging for improved stability. KnOTS (Stoica et al. 2024) introduces SVD-based parameter-space alignment to enable seamless fusion of conflicting LoRA adapters without retraining. Based on these methods, our approach employs task vectors as differentiable guidance signals to establish dynamic equilibrium among competing optimization objectives across tasks.

In-Context Learning

In-context learning has emerged as a novel paradigm in machine learning (Rubin, Herzig, and Berant 2021; Dong et al. 2022; Wies, Levine, and Shashua 2023). The foundational work by (Brown et al. 2020) first demonstrated the in-context learning capability of large language models (LLMs) when provided with a limited number of examples. Building upon this, MetaICL (Min et al. 2021) integrates tasks into the ICL format, enabling models to achieve performance comparable to direct fine-tuning without extensive parameter adjustments. LaMDA (Thoppilan et al. 2022) emphasizes the command tuning of the model for a better understanding of task descriptions. To address ICL's limitations in multi-step reasoning, (Wei et al. 2022) introduced Chain-of-Thought (CoT) prompting, which decomposes tasks into intermediate reasoning steps. Building on this, (Press et al. 2022) proposed Self-Ask, a hierarchical prompting framework that breaks complex queries into subquestions. The fusion of ICL with LoRA emerged as a breakthrough for balancing adaptability and efficiency. In scenario understanding, IC-LoRA (Huang et al. 2024) is a novel approach that leverages in-context learning to adapt models to various tasks without extensive fine-tuning. At the same time, ICM-LoRA (Shao et al. 2025) extends the concept of IC-LoRA by incorporating meta-learning and conditional variational autoencoders (CVAE). In this paper, our framework acquires task-specific adaptation vectors through in-context learning by modeling the relationship between LoRA weight perturbations and context demonstrations.

Meta Learning

Meta learning, or "learning to learn" (Brazdil et al. 2008), has garnered significant attention in the machine learning community (Vanschoren 2018; Finn, Abbeel, and Levine 2017; Li et al. 2017; Lee et al. 2019; Hsu, Levine, and Finn 2018). Early breakthroughs, such as model-agnostic meta-learning (MAML) (Finn, Abbeel, and Levine 2017), demonstrated that initializing model parameters in a task-agnostic basin enables rapid adaptation with a few gradient steps. In the context of few-shot learning, Matching Networks (Vinyals et al. 2016) utilizes a siamese network architecture to match new examples with previously seen ones. At the same time, Relationnet (Sung et al. 2018) proposes a two-branch relational network that performs a few-shot learning by learning to compare query images with a small number of labeled sample images. Recent advances have focused on improving the efficiency and scalability

of meta-learning. Developing first-order meta learning algorithms, such as FOMAML and Reptile (Nichol, Achiam, and Schulman 2018), has reduced the computational overhead associated with second-order derivatives in MAML. To develop robust LLMs adaptable to unseen tasks, meta-training approaches like MetaICL (Min et al. 2021) and MetaICT (Chen et al. 2021) have been proposed, involving meta-training pre-trained LLMs models on diverse tasks through in-context multi-task fine-tuning and evaluating on disjoint test sets. Building on these, MAML-en-LLM (Sinha et al. 2024) can learn truly generalizable parameters that perform well in disjoint tasks and adapt to unseen tasks. Our work aims to merge meta-learning principles with in-context learning in the context of LoRA adapters.

Methodology

Problem Settings

We focus on **In-Context Meta LoRA Fusion**, aiming to address the multitask fusion problem for LoRA fine-tuned models by fusing the latent spaces of task-specific Variational Autoencoders (VAEs). Consider n isomorphic models $\{f^{(1)}, f^{(2)}, \ldots, f^{(n)}\}$, each fine-tuned on a distinct task $\mathcal{T}_i$ (e.g., image classification, object detection) from a shared pre-trained checkpoint f^{pt}. The parameters of the pre-trained model are denoted as $\theta^{\text{pt}} = \{W_j^{\text{pt}}\}_{j=1}^{L}$, where W_j^{pt} represents the weights of the j-th layer and L is the total number of layers. Each task-specific LoRA model is associated with a VAE (VAE$_i$), which encodes task-specific LoRA parameters into a latent space, enabling cross-task fusion in this latent space. This approach aims to improve the recognition accuracy of the merged model by effectively integrating task-specific knowledge.

Task Vector Extraction

Following (Hendel, Geva, and Globerson 2023; Li et al. 2025a), we define a **task vector** $\Delta\mathbf{v}_{\mathcal{T}_i}$ for task $\mathcal{T}_i$ as the element-wise difference between the final-layer output tokens of the fine-tuned and pre-trained models:

$$\Delta\mathbf{v}_{\mathcal{T}_i} = \mathbf{z}^*_{\mathcal{T}_i} - \mathbf{z}^{(0)}, \tag{1}$$

where $\mathbf{z}^{(0)}$ denotes the output tokens from the last layer of the pre-trained model, and $\mathbf{z}^*_{\mathcal{T}_i}$ denotes the output tokens from the last layer after fine-tuning on task $\mathcal{T}_i$. The task vector $\Delta\mathbf{v}_{\mathcal{T}_i}$ thus captures the knowledge and adaptation required for the model to specialize in task $\mathcal{T}_i$ based on its final representations.For notational brevity, we use $\mathbf{v}_{\mathcal{T}i}$ to denote the task vector $\Delta\mathbf{v}\mathcal{T}_i$ as defined in Eq. 1.

Fusion VAE for Latent Space Encoding

We introduce a **Fusion VAE** method for encoding LoRA parameters in the latent space, enabling effective cross-task fusion. This approach differs from traditional CVAE methods in its handling of task semantics. Specifically, it represents each task's LoRA parameters $\mathbf{l}^{(i)} = \text{flatten}(\tau^{(i)}) \in \mathbb{R}^D$ in the latent space to facilitate subsequent fusion operations.

Let ϕ denote the parameters of the encoder (inference network), and θ denote the parameters of the decoder (generative network) in the VAE framework. Throughout this paper, we use θ_i to represent the LoRA parameters associated with task $\mathcal{T}_i$, and ψ to denote the decoder network parameters when necessary.

We employ a variational encoder to model the distribution over the latent variable $\mathbf{z}$ given both the LoRA parameters and the corresponding task vector. Specifically, we introduce an approximate posterior distribution $q_\phi(\mathbf{z} \mid \mathbf{l}^{(i)}, \mathbf{v}_{\mathcal{T}_i})$. This distribution is learned to approximate the true posterior $p(\mathbf{z} \mid \mathbf{l}^{(i)}, \mathbf{v}_{\mathcal{T}_i})$ over the latent variable $\mathbf{z}$ conditioned on the observed inputs.

The encoder of the Fusion VAE takes the concatenation of $\mathbf{l}^{(i)}$ and $\mathbf{v}_{\mathcal{T}_i}$ as input, and outputs the mean μ_ϕ and variance σ^2_ϕ of a Gaussian distribution in the latent space:

$$\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{l}^{(i)}, \mathbf{v}_{\mathcal{T}_i}) = \mathcal{N}\left(\mu_\phi([\mathbf{l}^{(i)}; \mathbf{v}_{\mathcal{T}_i}]),\, \sigma^2_\phi([\mathbf{l}^{(i)}; \mathbf{v}_{\mathcal{T}_i}])\mathbf{I}\right), \tag{2}$$

where $[\mathbf{l}^{(i)}; \mathbf{v}_{\mathcal{T}_i}]$ denotes the concatenation of the two input vectors, and $\mathbf{I}$ is the identity matrix that ensures the covariance is diagonal.

The decoder reconstructs the LoRA parameters from a latent vector $\mathbf{z} \in \mathbb{R}^k$ and $\mathbf{v}_{\mathcal{T}_i}$:

$$\hat{\mathbf{l}}^{(i)} \sim p_\theta(\mathbf{l}^{(i)} \mid \mathbf{z}, \mathbf{v}_{\mathcal{T}_i}) = \mathcal{N}\left(\hat{\mu}_\theta([\mathbf{z}; \mathbf{v}_{\mathcal{T}_i}]), \hat{\sigma}^2_\theta([\mathbf{z}; \mathbf{v}_{\mathcal{T}_i}])\mathbf{I}\right). \tag{3}$$

where $\hat{\mu}$ and $\hat{\sigma}$ denote the mean and variance outputs of the decoder network parameterized by θ, given the concatenated latent vector and task vector as input.

To balance reconstruction accuracy and latent space regularization, we maximize the Evidence Lower Bound (ELBO):

$$\mathcal{L}_i = \mathbb{E}_{q_\phi}\left[\log p_\theta(\mathbf{l}^{(i)} \mid \mathbf{z}, \mathbf{v}_{\mathcal{T}_i})\right] - \text{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{l}^{(i)}, \mathbf{v}_{\mathcal{T}_i}) \parallel \mathcal{N}(0, \mathbf{I})\right). \tag{4}$$

In the KL divergence term, $\mathcal{N}(0, \mathbf{I})$ denotes the standard normal prior distribution (i.e., mean zero and identity covariance), which regularizes the latent space towards a unit Gaussian.

In-Context Meta Learning

Traditional meta-learning often requires explicit task boundaries and repeated retraining, which is impractical for large models and dynamic task settings. In-context meta learning overcomes these limitations by leveraging contextual information (e.g., LoRA parameters and task vectors) at inference time, enabling rapid adaptation and flexible fusion without costly retraining. This approach improves scalability and generalization to new or evolving tasks by utilizing latent representations learned from previous experiences.

Our methodology focuses on an advanced meta-learning framework designed to optimize the integration of multiple task-specific VAEs. This framework consists of two main components: task-specific adaptation and meta-parameter updating.

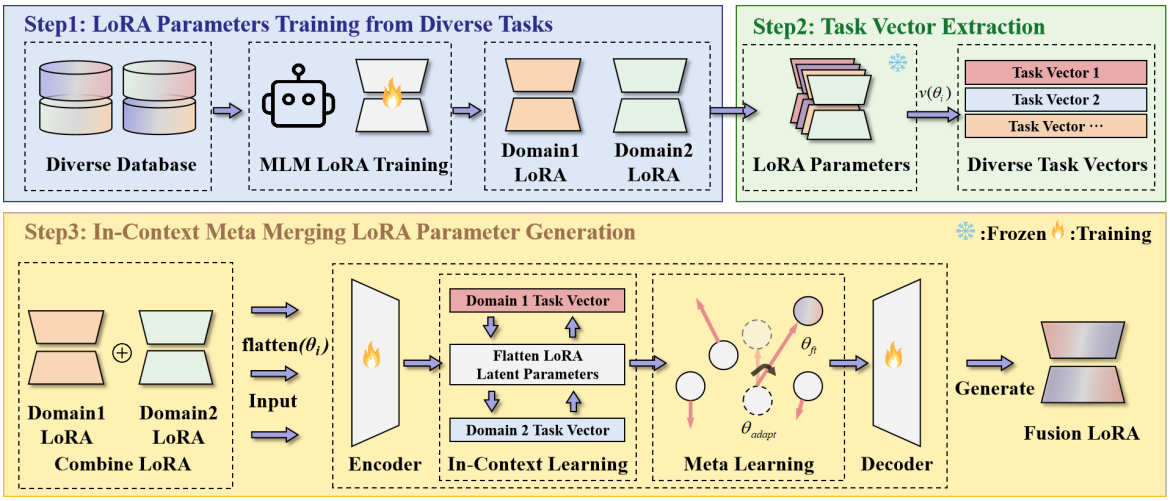

Figure 2: **Overview of the In-Context Meta Fusion LoRA framework. Step 1:** LoRA adapters are trained on a diverse set of tasks using masked language modeling (MLM), resulting in task-specific LoRA weights. **Step 2:** For each task, a task vector is obtained by computing the parameter difference between the fine-tuned and pre-trained LoRA adapters, providing a compact representation of task-specific knowledge. **Step 3:** The Fusion VAE leverages in-context learning and meta-learning, taking multiple task vectors and LoRA parameters as input to model latent task relationships. For each parameter dimension, the input θ_{adapt} refers to the value of that specific dimensions relevant to the task in the flattened LoRA parameters; the VAE then adjusts θ_{adapt} to generat e the task-specialized parameter θ_{ft} suitable for the target task. The encoder-decoder structure enables efficient parameter fusion and adaptation.

During the task-specific adaptation phase, each task-specific VAE is adapted to its corresponding task. The encoder E_ϕ processes the task-specific LoRA parameters θ_i and generates a latent distribution, which is defined by its mean μ_i and logarithmic variance $\log \sigma_i^2$:

$$(\mu_i, \log \sigma_i^2) = E_\phi(\theta_i). \tag{5}$$

From this latent distribution, a sample vector $\mathbf{z}_i \in \mathbb{R}^d$ is drawn, where d denotes the dimensionality of the latent space. This sample is then used by the decoder D_ψ to generate an initial parameter estimate θ_i^{init}:

$$\theta_i^{\text{init}} = D_\psi(\mathbf{z}_i). \tag{6}$$

This initial estimate is refined through K steps of gradient-based adaptation to minimize the task-specific reconstruction loss:

$$\theta_i^{(0)} = \theta_i^{\text{init}} \tag{7}$$

$$\theta_i^{(k)} = \theta_i^{(k-1)} - \beta \nabla_\theta \mathcal{L}_{\text{task}}(\theta_i^{(k-1)}, \mathcal{D}_i), \tag{8}$$

where $k = 1, \ldots, K$.

The final adapted parameter is denoted as:

$$\theta_i^{\text{adapt}} = \theta_i^{(K)} \tag{9}$$

In the meta-parameter updating component, the focus is on enhancing the fusion capabilities of the VAE parameters. The reconstruction loss is calculated to evaluate how accurately the original parameters can be reconstructed from the adapted latent representation:

$$\mathcal{L}_{\text{recon}} = \text{MSE}(\theta_i, \theta_i^{\text{recon}}). \tag{10}$$

This loss measures the discrepancy between the original and reconstructed parameters. Additionally, the KL divergence loss is computed to regularize the latent distribution, ensuring it aligns with a standard normal prior:

$$\mathcal{L}_{\text{KL}} = -\frac{1}{2} \sum \left(1 + \log \sigma_i^{\text{adapt}^2} - (\mu_i^{\text{adapt}})^2 - \sigma_i^{\text{adapt}^2}\right). \tag{11}$$

This loss encourages the latent space to maintain a structure that is consistent with a normal distribution, which is beneficial for generating new data.

These two losses are combined into a meta-loss, which guides the optimization of the VAE parameters:

$$\mathcal{L}_{\text{meta}} = \mathcal{L}_{\text{recon}} + \lambda_{\text{KL}} \cdot \mathcal{L}_{\text{KL}}, \tag{12}$$

where $\lambda_{\text{KL}} > 0$ is a weighting coefficient that balances the regularization term (KL divergence) and the reconstruction objective. This formulation ensures that the learned latent space is both informative for reconstruction and regularized to match the prior distribution, which is essential for robust task fusion.

In particular, the meta-loss in Eq. 12 shares a close connection with the evidence lower bound (ELBO) in Eq. 4, which forms the theoretical foundation of variational autoencoders. While Eq. 4 describes the objective for reconstructing the original LoRA parameters for a single task, Eq. 12 extends this principle to the meta-learning setting, where the optimization is performed over multiple tasks to promote generalization and effective fusion across the task batch.

The encoder and decoder parameters are updated based

Algorithm 1: In-Context Meta LoRA Fusion Meta-Learning Procedure

Require: Task set $\mathcal{T}$, initial VAE parameters ϕ, ψ, meta step size γ, adaptation step size β, KL weight λ_{KL}

for $\text{metaIter} = 1$ to M **do**
  Sample a batch of tasks $\{\mathcal{T}_i\}$ from $\mathcal{T}$
  for each task $\mathcal{T}_i$ in the batch **do**
    $\theta_i \leftarrow \text{ObtainLoRAParams}(\mathcal{T}_i)$
    $v_{\mathcal{T}_i} \leftarrow \text{ComputeTaskVector}(\mathcal{T}_i)$ /* Eq. (1) */
    $(\mu_i, \log \sigma_i^2) \leftarrow E_\phi([\theta_i; v_{\mathcal{T}_i}])$
    $z_i \sim \mathcal{N}\big(\mu_i, \text{diag}(\sigma_i^2)\big)$
    $\theta_i^{\text{init}} \leftarrow D_\psi([z_i; v_{\mathcal{T}_i}])$
    $\theta_i^{\text{adapt}} \leftarrow \theta_i^{\text{init}} - \beta \nabla_{\theta_i^{\text{init}}} \mathcal{L}_{\text{task}}\big(\theta_i^{\text{init}}, \mathcal{D}_i\big)$
    $(\mu_i', \log \sigma_i'^2) \leftarrow E_\phi([\theta_i^{\text{adapt}}; v_{\mathcal{T}_i}])$
    $\theta_i^{\text{recon}} \leftarrow D_\psi([z_i; v_{\mathcal{T}_i}])$
    $\mathcal{L}_{\text{recon}} \leftarrow \text{MSE}(\theta_i, \theta_i^{\text{recon}})$
    $\mathcal{L}_{\text{KL}} \leftarrow \text{KL}\Big(\mathcal{N}(\mu_i', \sigma_i'^2), \mathcal{N}(0, I)\Big)$ /* Eq. (11) */
    $\mathcal{L}_{\text{meta}} \leftarrow \mathcal{L}_{\text{recon}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}$
  end for
  $\phi \leftarrow \phi - \gamma \nabla_\phi \Big(\sum_i \mathcal{L}_{\text{meta}}\Big)$
  $\psi \leftarrow \psi - \gamma \nabla_\psi \Big(\sum_i \mathcal{L}_{\text{meta}}\Big)$
end for

on this meta-loss:

$$\phi \leftarrow \phi - \gamma \nabla_\phi \mathcal{L}_{\text{meta}}, \tag{13}$$

$$\psi \leftarrow \psi - \gamma \nabla_\psi \mathcal{L}_{\text{meta}}. \tag{14}$$

Here, γ denotes the meta step size used for updating the VAE parameters (ϕ, ψ) across tasks, in contrast to the inner-loop adaptation step size β in Eq. 8, which controls the gradient-based update for task-specific parameter adaptation. This separation ensures that meta-learning proceeds at an appropriate scale, independent of the rapid adaptation dynamics within each task.

This meta-learning process enables the Fusion VAE to effectively assimilate knowledge from multiple tasks, thereby significantly enhancing the performance of the merged model across various recognition tasks. By iteratively adapting to individual tasks and updating the model based on a meta-loss, the VAE develops a robust and generalizable representation of the latent space. This representation captures the essential features of each task while maintaining the model's ability to merge them effectively.

Experiments

In this section, we discuss the experimental setup and implementation details. Additionally, we report the performance of different models on vision and language tasks.

Experiment Settings

Baselines. We adopted the original model with LoRA (Hu et al. 2022), which is generated by Model Soup (Wortsman et al. 2022a), RegMean (Jin et al. 2022), TA (Ilharco et al. 2022), KnOTS (Stoica et al. 2024), TIES (Yadav et al. 2023), and DARE (Yu et al. 2024) as baselines to compare with our method, aiming to assess its advantages on various tasks.

Datasets. For the computer vision task, we focus on the target detection task and the multimodal question-answering task. So we employ the VOC 2012 dataset (Everingham et al. 2010) and the ScienceQA dataset (Lu et al. 2022). For the language task, we utilize The Pile (Gao et al. 2020) as the language modeling task dataset.

Data Preparation. During model fine-tuning, a sequence of LoRA matrices $\{L_t\}_{t=1}^{T}$ with varying ranks r is generated for multiple diverse tasks, where T denotes the number of fine-tuning steps. Each matrix $L_t \in \mathbb{R}^{m \times n}$ is flattened into a one-dimensional array $l_t \in \mathbb{R}^{m \times n}$ to enable alignment with the task vector v_{task}. These flattened LoRA parameters, combined with their corresponding task vectors, form the training data set $\{(v_{\text{task}}, l_t)\}$ for the **self-designed VAE**. For the selected tasks, we obtain their respective LoRA parameter sequences $\{L_{t1}\}_{t=1}^{T1}$ and $\{L_{t2}\}_{t=1}^{T2}$, flatten them into arrays l_{t1} and l_{t2}, and construct data sets $D_1 = \{(v_{\text{task1}}, l_{t1})\}$ and $D_2 = \{(v_{\text{task2}}, l_{t2})\}$. These data sets are merged into a combined data set $D_{\text{combined}} = D_1 \cup D_2$. During VAE training, batches are randomly drawn from D_{combined}, enabling the VAE to learn from the different LoRA parameters of both tasks and forming the basis for subsequent parameter generation and model customization.

Main Results

We conducted experiments on image object detection on Florence2 (Xiao et al. 2024), visual question answering on LLAVA-v1.5 (Liu et al. 2024), and language modeling tasks on Llama3 (Grattafiori et al. 2024) to demonstrate the reliability and generalizability of our method.

Results on Vision Tasks. As shown in Table 1, we compare various model fusion methods on object detection tasks with multiple categories. Our ICM-Fusion achieves the highest MAP50 scores and shows improvement in the relevant metrics. Although detection performance decreases in some samples, the overall results remain superior to existing methods. This indicates that ICM-Fusion can enhance the capability of the fused model and optimize its performance when task vectors point in similar directions. Although affected by opposing vectors, ICM-Fusion effectively mitigates their negative impact.

As shown in Table 2, For the visual question answering (VQA) task, while fine-tuned model scores may vary due to non-robust benchmark evaluations, ICM-Fusion achieves a consistent and significant improvement in the average score across all tasks post-fusion compared to existing methods. This result demonstrates its capability to enhance the reasoning performance of merged models. Moreover, since LLaVA-v1.5 (Liu et al. 2024) fine-tunes only the decoder while Florence-2 (Xiao et al. 2024) requires fine-tuning both the encoder and decoder, there are certain structural differences between the fused models. Both tasks demonstrate that ICM-Fusion can robustly improve the performance of the merged model, indicating that ICM-Fusion is adaptable to LoRA-based fusion of models with different architectures.

Few-shot Results. Table 3 reports the results in few-shot settings. We evaluated MAP50 for several long-tail cate-

Method	MAP50								
	Cat	Dog	Train	Bus	Bicycle	Horse	Sheep	Bird	Avg.
Original Model	0.94	0.92	0.89	0.87	0.85	0.83	0.81	0.76	0.86
Original LoRA	0.95	0.92	0.91	**0.92**	0.88	**0.90**	0.85	0.82	0.89
Model Soup (Wortsman et al. 2022a)	0.93	0.92	0.88	0.84	0.85	0.85	0.82	0.75	0.84
RegMean (Jin et al. 2022)	0.95	0.92	0.90	0.89	0.87	0.86	0.85	0.82	0.88
TA (Ilharco et al. 2022)	0.95	0.93	0.91	0.90	0.88	0.87	0.86	0.83	0.89
SVD in Latent State (Koren 2008)	0.89	0.87	0.85	0.84	0.84	0.81	0.80	0.73	0.83
KnOTS (Stoica et al. 2024)	0.94	0.92	0.90	0.89	0.86	0.83	0.82	0.76	0.87
TIES (Yadav et al. 2023)	0.93	0.91	0.88	0.87	0.85	0.83	0.81	0.77	0.86
DARE-TIES (Yu et al. 2024)	0.92	0.90	0.87	0.86	0.84	0.82	0.80	0.76	0.85
ICM-Fusion (Ours)	**0.96**	**0.93**	**0.92**	0.91	**0.90**	0.88	**0.87**	**0.85**	**0.90**

Table 1: **Results on object detection with rank $r = 8$.** Compared to the original LoRA, ICM-Fusion achieves significant improvement in most samples and demonstrates enhanced overall detection performance.

Method	NAT	SOC	LAN	TXT	IMG	NO	Avg.
Original Model	89.32	96.08	85.32	88.95	87.17	88.43	89.38
Original LoRA	**89.34**	**96.10**	85.30	88.94	87.16	88.44	89.38
Model Soup (Wortsman et al. 2022a)	87.61	93.02	85.33	82.91	86.83	85.23	86.82
RegMean (Jin et al. 2022)	88.91	95.82	85.11	88.42	87.08	88.01	87.73
TA (Ilharco et al. 2022)	89.01	95.96	85.21	88.70	87.12	88.15	87.86
SVD (Koren 2008)	88.85	95.68	85.18	88.31	87.03	88.05	87.85
KnOTS (Stoica et al. 2024)	89.10	96.03	85.27	88.81	87.16	88.34	88.12
TIES (Yadav et al. 2023)	88.98	95.94	85.24	88.65	87.14	88.22	88.03
DARE-TIES (Yu et al. 2024)	89.12	96.05	85.29	88.83	87.16	88.38	88.14
ICM-Fusion (Ours)	89.33	96.08	85.32	**88.96**	**87.17**	**88.45**	**89.39**

Table 2: **Result on ScienceQA.** Question classes: NAT = natural science, SOC = social science, LAN = language science, TXT = text context, IMG = image context, NO = no context.

gories, Sofa, Airplane, Motorbike, and Dining Table, under two data regimes: 0% (no additional training data) and +10% (with a small amount of additional data). Across all methods, performance is limited when no extra data is provided. However, when 10% more data is available, all methods exhibit improvements, with ICM-Fusion demonstrating the most significant gains and achieving the highest MAP50 scores in the few-shot category. This demonstrates that ICM-Fusion can still generalize to the target domain even in few-shot learning scenarios.

Results on Language Tasks. To validate the generalization capability of ICM-Fusion across different modal tasks, we selected a pure language modeling task with LLAMA3 (Grattafiori et al. 2024) on the Pile dataset (Gao et al. 2020). To assess the effectiveness of ICM-Fusion in the context of language modeling, we conduct experiments across multiple established benchmarks. For consistency, we also conducted experiments on language modeling tasks using related methods, as shown in Table 4. For most samples, the PPL shows a decrease compared to the original LoRA before fusion, indicating that the fused LoRA effectively enhances the language modeling capability of the model. Although the BPC change before and after fine-tuning is relatively small, the fused model still achieves better BPC performance on most samples, demonstrating that ICM-Fusion adapts well to language modeling tasks.

Based on the results from Tables 1, 2, and 4, ICM-Fusion demonstrates adaptability to fine-tune LoRA across different model architectures and multimodal tasks. This indicates that our method not only enhances the capabilities of pre-fusion domain-specific LoRA but also generalizes effectively to diverse models and cross-modal applications.

These results highlight the robustness and adaptability of ICM-Fusion in few-shot scenarios, particularly for rare classes where training samples are scarce. The substantial improvement over baseline fusion methods underscores its effectiveness for long-tail recognition under limited data conditions.

Ablation Study

To investigate the impact of source data hyperparameters on the performance of ICM-Fusion, we conducted ablation studies on the rank of LoRA (which determines the number of LoRA parameters) and the proportion of sampled data.

Effect of LoRA Ranks. Table 5 presents the results of varying the LoRA rank r in ICM-Fusion. We observe that MAP50 scores for representative categories (Cat, Dog, Bus) remain consistently high across different ranks. The results demonstrate that the performance of the fused model remains nearly constant across different LoRA ranks (pa-

Method	Sofa		Aeroplane		Motorbike		Dining Table	
	0%	+10%	0%	+10%	0%	+10%	0%	+10%
Original Model	0.00	0.79	0.00	0.75	0.00	0.73	0.00	0.76
Original LoRA(Cat)	0.00	0.83	0.00	0.77	0.00	0.75	0.00	0.81
Model Soup (Wortsman et al. 2022a)	0.00	0.78	0.00	0.74	0.00	0.72	0.00	0.78
RegMean (Jin et al. 2022)	0.00	0.81	0.00	0.76	0.00	0.74	0.00	0.80
TA (Ilharco et al. 2022)	0.00	0.82	0.00	0.77	0.00	0.75	0.00	0.81
SVD in Latent State (Koren 2008)	0.00	0.77	0.00	0.73	0.00	0.71	0.00	0.78
KnOTS (Stoica et al. 2024)	0.00	0.80	0.00	0.75	0.00	0.73	0.00	0.80
TIES (Yadav et al. 2023)	0.00	0.79	0.00	0.74	0.00	0.72	0.00	0.79
DARE-TIES (Yu et al. 2024)	0.00	0.78	0.00	0.74	0.00	0.72	0.00	0.78
CM-Fusion (Ours)I	0.00	0.85	0.00	0.80	0.00	0.79	0.00	0.85

Table 3: **MAP50 scores for long-tail classes.**For different model fusion pipelines under two data percentages.

Method	ArXiv		Books		Ubuntu		Wikipedia		Gutenberg		Avg.	
	PPL	BPC	PPL	BPC	PPL	BPC	PPL	BPC	PPL	BPC	PPL	BPC
Original Model	7.76	0.44	7.67	0.51	10.00	0.59	6.07	0.46	8.75	0.63	8.45	0.53
Original LoRA	6.70	0.41	7.10	0.48	9.67	0.59	5.58	0.44	8.60	0.62	7.53	0.51
Model Soup (Wortsman et al. 2022a)	7.00	0.43	7.08	0.48	9.67	0.59	5.56	0.45	8.61	0.63	7.58	0.52
RegMean (Jin et al. 2022)	6.90	0.43	7.10	0.48	9.68	0.59	5.60	0.45	8.63	0.62	7.58	0.51
TA (Ilharco et al. 2022)	6.85	0.43	7.09	0.48	9.67	0.59	5.57	0.45	8.62	0.62	7.56	0.51
SVD in Latent State (Koren 2008)	6.84	0.43	7.13	0.48	9.69	0.59	5.60	0.45	8.66	0.63	7.58	0.52
KnOTS (Stoica et al. 2024)	6.78	0.43	7.13	0.48	9.70	0.58	5.61	0.45	8.68	0.62	7.58	0.51
TIES (Yadav et al. 2023)	6.79	0.43	7.14	0.48	9.71	0.59	5.62	0.45	8.69	0.63	7.59	0.52
DARE-TIES (Yu et al. 2024)	6.81	0.43	7.16	0.48	9.72	0.58	5.63	0.45	8.70	0.62	7.60	0.51
ICM-Fusion (Ours)	**6.73**	**0.42**	**7.07**	**0.47**	**9.64**	**0.58**	**5.55**	**0.43**	**8.57**	**0.61**	**7.51**	**0.50**

Table 4: Language modeling results on different datasets (ArXiv, Books, Ubuntu, Wikipedia, Gutenberg) for various model fusion methods. Metrics reported are Perplexity (PPL) and Bits-per-Character (BPC), and their average values.

rameter numbers). Thus, ICM-Fusion exhibits robustness to LoRA models with varying parameter numbers.

Rank	Parameter	ICM-Fusion		
		Cat	Dog	Bus
$r = 1$	241,241	0.96	0.93	0.91
$r = 2$	482,482	0.96	0.93	0.91
$r = 4$	964,964	0.96	0.93	0.91
$r = 8$	1,929,928	0.96	0.93	0.91
$r = 16$	3,859,856	0.96	0.93	0.91

Table 5: **Impact of LoRA Rank and Parameter.** ICM-Fusion demonstrates robust performance across LoRA weights with different ranks and parameters.

Effect of Data Sampling Rate. As shown in Table 6, we further analyze the sensitivity of ICM-Fusion to the amount of available training data for several long-tail classes. With no additional data, MAP50 scores are near zero. However, as the proportion of training data increases from 10% to 30%, substantial and consistent improvements are observed across all long-tail classes. These results highlight the data efficiency of ICM-Fusion and its ability to benefit from even small increases in training samples rapidly. Therefore, even in scenarios with limited training samples, ICM-Fusion can still achieve domain enhancement.

Data Percent	Sofa	Aeroplane	Motorbike	Dining Table
0%	0.00	0.00	0.00	0.00
+10%	0.83	0.79	0.81	0.80
+20%	0.85	0.82	0.83	0.83
+30%	0.88	0.85	0.87	0.86

Table 6: MAP50 scores for long-tail classes under different proportions of training data.

Conclusion

We propose a unified framework for In-Context Meta LoRA Fusion, which systematically integrates task-specific LoRA parameters via a Fusion VAE and meta-learning. ICM-Fusion generates an optimal fused model architecture by performing meta-learning-guided task vector direction optimization in the latent space after VAE sampling. Our experiments demonstrate that the method not only reduces storage overhead but also adapts effectively across vision and language tasks, even when direct supervision or extensive training are not readily available. We further discuss potential applications, such as sharing knowledge across multiple domains and augmenting new tasks with minimal overhead, thereby enhancing robustness and versatility in multi-task systems. We believe our empirical investigation offers deeper insights into parameter-efficient model unification.

Acknowledgments

This research was supported by funding from the Hong Kong RGC General Research Fund (Nos. 15221123 and 15216424), the PolyU Internal Fund (No. P0058468), and the Huawei Gifted Fund.

References

Brazdil, P.; Carrier, C. G.; Soares, C.; and Vilalta, R. 2008. *Metalearning: Applications to data mining*. Springer science & business media.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Caruana, R. 1997. Multitask learning. *Machine learning*, 28(1): 41–75.

Chen, Y.; Zhong, R.; Zha, S.; Karypis, G.; and He, H. 2021. Meta-learning via language model in-context tuning. *arXiv preprint arXiv:2110.07814*.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Ma, J.; Li, R.; Xia, H.; Xu, J.; Wu, Z.; Liu, T.; et al. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. *http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Gao, L.; Biderman, S.; Black, S.; Golding, L.; Hoppe, T.; Foster, C.; Phang, J.; He, H.; Thite, A.; Nabeshima, N.; et al. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *arXiv preprint arXiv:2101.00027*.

Gepperth, A.; and Hammer, B. 2016. Incremental learning algorithms and applications. In *European symposium on artificial neural networks (ESANN)*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hendel, R.; Geva, M.; and Globerson, A. 2023. In-context learning creates task vectors. *arXiv preprint arXiv:2310.15916*.

Hospedales, T.; Antoniou, A.; Micaelli, P.; and Storkey, A. 2021. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5149–5169.

Hsu, K.; Levine, S.; and Finn, C. 2018. Unsupervised learning via meta-learning. *arXiv preprint arXiv:1810.02334*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Huang, L.; Wang, W.; Wu, Z.-F.; Shi, Y.; Dou, H.; Liang, C.; Feng, Y.; Liu, Y.; and Zhou, J. 2024. In-context lora for diffusion transformers. *arXiv preprint arXiv:2410.23775*.

Huang, Y.; Lin, Z.; Liu, H.; et al. 2023. ComPEFT: Computationally Efficient LoRA Fusion via Shared Sparse Adapters. In *EMNLP*.

Ilharco, G.; Ribeiro, M. T.; Wortsman, M.; Gururangan, S.; Schmidt, L.; Hajishirzi, H.; and Farhadi, A. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Jang, D.-H.; Yun, S.; and Han, D. 2024. Model stock: All we need is just a few fine-tuned models. In *European Conference on Computer Vision*, 207–223. Springer.

Jiang, W.; Li, D.; Hu, M.; Zhai, G.; Yang, X.; and Zhang, X.-P. 2024. Few-shot class-incremental learning with prior knowledge. *arXiv preprint arXiv:2402.01201*.

Jin, X.; Ren, X.; Preotiuc-Pietro, D.; and Cheng, P. 2022. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*.

Kingma, D. P.; Welling, M.; et al. 2013. Auto-encoding variational bayes.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 426–434.

Lee, K.; Maji, S.; Ravichandran, A.; and Soatto, S. 2019. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10657–10665.

Li, H.; Zhang, Y.; Zhang, S.; Chen, P.-Y.; Liu, S.; and Wang, M. 2025a. When is Task Vector Provably Effective for Model Editing? A Generalization Analysis of Nonlinear Transformers. In *The Thirteenth International Conference on Learning Representations*.

Li, W.; Peng, Y.; Zhang, M.; Ding, L.; Hu, H.; and Shen, L. 2023. Deep model fusion: A survey. *arXiv preprint arXiv:2309.15698*.

Li, W.; Zou, L.; Tang, M.; Yu, Q.; Li, W.; and Li, C. 2025b. META-LORA: Memory-Efficient Sample Reweighting for Fine-Tuning Large Language Models. In *Proceedings of the 31st International Conference on Computational Linguistics*, 8504–8517.

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 26296–26306.

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35: 2507–2521.

Masana, M.; Liu, X.; Twardowski, B.; Menta, M.; Bagdanov, A. D.; and Van De Weijer, J. 2022. Class-incremental

learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5): 5513–5533.

Min, S.; Lewis, M.; Zettlemoyer, L.; and Hajishirzi, H. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Press, O.; Zhang, M.; Min, S.; Schmidt, L.; Smith, N. A.; and Lewis, M. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Rubin, O.; Herzig, J.; and Berant, J. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Shao, Y.; Liang, S.; Lin, X.; Ling, Z.; Zhu, Z.; Yan, M.; Liu, H.; Chen, S.; Yan, Z.; Meng, Y.; et al. 2024. GWQ: Gradient-Aware Weight Quantization for Large Language Models. *arXiv preprint arXiv:2411.00850*.

Shao, Y.; Yan, M.; Liu, Y.; Chen, S.; Chen, W.; Long, X.; Yan, Z.; Li, L.; Zhang, C.; Sebe, N.; et al. 2025. In-Context Meta LoRA Generation. *arXiv preprint arXiv:2501.17635*.

Sinha, S.; Yue, Y.; Soto, V.; Kulkarni, M.; Lu, J.; and Zhang, A. 2024. Maml-en-llm: Model agnostic meta-training of llms for improved in-context learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2711–2720.

Stoica, G.; Ramesh, P.; Ecsedi, B.; Hoffman, J.; and Choshen, L. 2024. KnOTS: Model Merging with SVD to Tie the Knots. *arXiv preprint arXiv:2410.19735*.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1199–1208.

Tang, A.; Shen, L.; Luo, Y.; Hu, H.; Du, B.; and Tao, D. 2024. Fusionbench: A comprehensive benchmark of deep model fusion. *arXiv preprint arXiv:2406.03280*.

Thoppilan, R.; De Freitas, D.; Hall, J.; Shazeer, N.; Kulshreshtha, A.; Cheng, H.-T.; Jin, A.; Bos, T.; Baker, L.; Du, Y.; et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Vandenhende, S.; Georgoulis, S.; Van Gansbeke, W.; Proesmans, M.; Dai, D.; and Van Gool, L. 2021. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7): 3614–3633.

Vanschoren, J. 2018. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Wang, H.; Ping, B.; Wang, S.; Han, X.; Chen, Y.; Liu, Z.; and Sun, M. 2024. Lora-flow: Dynamic lora fusion for large language models in generative tasks. *arXiv preprint arXiv:2402.11455*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wies, N.; Levine, Y.; and Shashua, A. 2023. The learnability of in-context learning. *Advances in Neural Information Processing Systems*, 36: 36637–36651.

Wortsman, M.; Ilharco, G.; Gadre, S. Y.; Roelofs, R.; Gontijo-Lopes, R.; Morcos, A. S.; Namkoong, H.; Farhadi, A.; Carmon, Y.; Kornblith, S.; et al. 2022a. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, 23965–23998. PMLR.

Wortsman, M.; Ilharco, G.; Kim, J. W.; Li, M.; Kornblith, S.; Roelofs, R.; Lopes, R. G.; Hajishirzi, H.; Farhadi, A.; Namkoong, H.; et al. 2022b. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7959–7971.

Wu, X.; Huang, S.; and Wei, F. 2024. Mixture of lora experts. *arXiv preprint arXiv:2404.13628*.

Wu, Y.; Huang, L.-K.; Wang, R.; Meng, D.; and Wei, Y. 2024. Meta continual learning revisited: Implicitly enhancing online hessian approximation via variance reduction. In *The Twelfth international conference on learning representations*, volume 2.

Xiao, B.; Wu, H.; Xu, W.; Dai, X.; Hu, H.; Lu, Y.; Zeng, M.; Liu, C.; and Yuan, L. 2024. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4818–4829.

Yadav, P.; Tam, D.; Choshen, L.; Raffel, C. A.; and Bansal, M. 2023. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36: 7093–7115.

Yu, L.; Yu, B.; Yu, H.; Huang, F.; and Li, Y. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Zhang, Q.; Chen, M.; Bukharin, A.; He, P.; Cheng, Y.; Chen, W.; and Zhao, T. 2023. AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. In *ICML*.

Zhang, W.; Deng, L.; Zhang, L.; and Wu, D. 2022. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2): 305–329.

Zhang, Y.; and Li, R. 2024. Dlp-lora: Efficient task-specific lora fusion with a dynamic, lightweight plugin for large language models. *arXiv preprint arXiv:2410.01497*.

Zoph, B.; Bello, I.; Kumar, S.; Du, N.; Huang, Y.; Dean, J.; Shazeer, N.; and Fedus, W. 2022. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*.