Neural Parameter Search for Slimmer Fine-Tuned Models and Better Transfer

Guodong Du[1] **Zitao Fang**[2] **Jing Li**[1]✉ **Junlin Li**[1] **Runhua Jiang**[2]
Shuyang Yu[2] **Yifei Guo**[2] **Yangneng Chen**[1] **Sim Kuan Goh**[2]
Ho-Kin Tang[1] **Daojing He**[1] **Honghai Liu**[1] **Min Zhang**[1]
[1]Harbin Institute of Technology, Shenzhen, China
[2]Xiamen University Malaysia
duguodong7@gmail.com jingli.phd@hotmail.com

Abstract

Foundation models and their checkpoints have significantly advanced deep learning, boosting performance across various applications. However, fine-tuned models often struggle outside their specific domains and exhibit considerable redundancy. Recent studies suggest that combining a pruned fine-tuned model with the original pre-trained model can mitigate forgetting, reduce interference when merging model parameters across tasks, and improve compression efficiency. In this context, developing an effective pruning strategy for fine-tuned models is crucial. Leveraging the advantages of the task vector mechanism, we preprocess fine-tuned models by calculating the differences between them and the original model. Recognizing that different task vector subspaces contribute variably to model performance, we introduce a novel method called **N**eural **P**arameter **S**earch (**NPS**) for slimming down fine-tuned models. This method enhances pruning efficiency by searching through neural parameters of task vectors within low-rank subspaces. Our method has three key applications: enhancing knowledge transfer through pairwise model interpolation, facilitating effective knowledge fusion via model merging, and enabling the deployment of compressed models that retain near-original performance while significantly reducing storage costs. Extensive experiments across vision, NLP, and multi-modal benchmarks demonstrate the effectiveness and robustness of our approach, resulting in substantial performance gains. The code is publicly available at: https://github.com/duguodong7/NPS-Pruning.

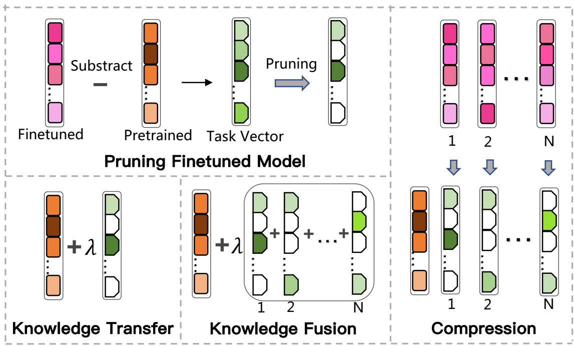

Figure 1: Knowledge transfer, fusion, and compression are enhanced with the assistance of pre-trained model parameters. The fine-tuned model is effectively represented as a combination of the pre-trained model and pruned task vectors, leading to knowledge retention.

1 Introduction

In recent years, with the release of foundational models and the proliferation of associated checkpoints, the field of machine learning has undergone a paradigm shift. This shift has significantly enhanced the performance of downstream applications. While fine-tuning pre-trained models (Wortsman et al., 2022; Choshen et al., 2022; Liu et al., 2022a) has become common practice, these models often struggle with generalization and perform poorly outside their specific domains. Consequently, improving knowledge transfer from pre-trained to fine-tuned models has become a recent research focus (Devlin et al., 2018). Consequently, recent research has increasingly focused on improving knowledge transfer, fusion, and compression by leveraging the parameters of the initial pre-trained model. Model Tailor (Zhu et al., 2024) prunes fine-tuned models and combines them with the original model to reduce catastrophic forgetting. Additionally, TALL-masks (Wang et al., 2024) compresses checkpoints by localizing task information within task vectors. All these research efforts on knowledge transfer with available pre-trained parameters depend on a crucial preprocessing step: pruning the fine-tuned model task vectors (Ilharco et al., 2023b), as shown in Figure 1.

Fine-tuned models often exhibit significant redundancy in parameter modifications compared to pre-trained models. Pruning these models can en-

✉ Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 32668–32687
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

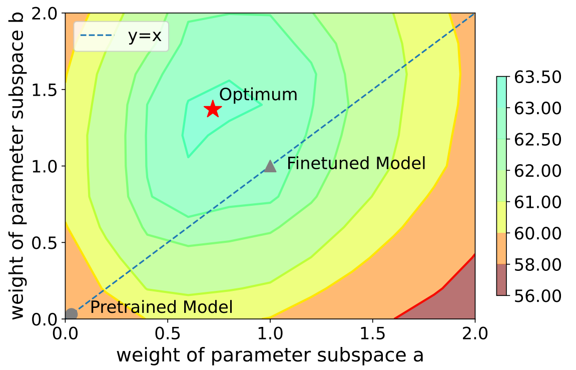

Figure 2: Performance of ViT-B/32 models on a specific task (SUN397 dataset). Different subspaces of neural parameters within the task vector contribute differently to the performance of the fine-tuned model.

hance the efficiency of knowledge representation. Pruning fine-tuned model sets offers three main advantages: First, it reduces conflicts between fine-tuned models and the pre-trained model during knowledge transfer, thereby enhancing resilience to catastrophic forgetting. Second, it minimizes interference among fine-tuned models during fusion, improving multi-task generalization capabilities. Finally, pruning finetuned models can reduce storage costs while maintaining multi-task performance. However, despite extensive research on model pruning in the context of compression (Liang et al., 2021; Yu et al., 2023a; Xia et al., 2022), there is a relative scarcity of studies focused specifically on pruning fine-tuned models. To address this gap, we propose a novel method called **N**eural **P**arameter **S**earch (**`NPS`**) and design an adapted approach to apply pruned fine-tuned models in scenarios such as knowledge transfer, fusion, and compression. Specifically, we leverage the advantages of the task vector mechanism and preprocess fine-tuned models by calculating the difference between them and the original model. Recognizing that different task vector subspaces contribute variably to model performance, as shown in Figure 2, we search through the neural parameters within low-rank subspaces of task vectors. We partition the fine-tuned parameters into a set number of subspaces based on their magnitude, use evolutionary algorithms to assign new weights to different subspaces, and update the weights based on the model's performance on calibration datasets. This process avoids the need for gradient calculations, offering lightweight and efficient advantages.

We validated the effectiveness of our method across three key applications: knowledge transfer, model fusion, and compression. First, we performed interpolation between NPS-pruned models and the pre-trained model to mitigate forgetting, demonstrating superior performance on the multi-modal benchmark with LLaVa (Zhu et al., 2024) model compared to previous methods. Second, we showed that weight averaging of multiple NPS-compressed fine-tuned models enables effective model fusion. Our approach was evaluated on NLP and vision tasks using models like T5 (Raffel et al., 2020), ViT (Dosovitskiy et al., 2020), and LLaMa2 (Touvron et al., 2023), as well as for fusing multiple PEFT adapters. Notably, it achieved a 4.3% performance gain with T5-base. Finally, for deployment, our method allowed compressed models to retain near-original fine-tuned performance while significantly reducing storage costs. Extensive experiments demonstrated a 40% improvement in compression efficiency on vision tasks.

Our **contributions** can be summarized in the following four points:

- We reveal the importance of pruning fine-tuned models and highlight the limitations of previous methods.
- We propose **N**eural **P**arameter **S**earch (**`NPS`**) for slimming down fine-tuned models.
- Based on the pruned fine-tuned models, we provide a simple and versatile method suitable for multi-task model fusion, compression, and robust knowledge transfer.
- Experimental results shown that our method significantly improves performance in various knowledge transfer scenarios.

2 Related Work

2.1 Knowledge Transfer, Fusion and Compression

In the realms of knowledge transfer, model fusion (Jiang et al., 2024; Fang et al., 2025), and compression, foundational studies have driven significant progress. (Wortsman et al., 2022) enhanced zero-shot learning by fine-tuning pre-trained models with minimal data, while (Houlsby et al., 2019) improved resource efficiency through parameter-efficient transfer learning. (Chen et al., 2020) advanced model compression and fusion using contrastive learning in unsupervised settings, collectively marking major strides in model efficiency and robustness.

Recent years have seen the emergence of innovative methods for enhancing performance and efficiency across tasks when both pre-trained and fine-tuned models are available. Fisher-weighted averaging (Matena and Raffel, 2022) uses an information-theoretic approach to assess parameter importance, while RegMean (Jin et al., 2022) offers a closed-form solution for merging parameters through local linear regression. Task Arithmetic (Ilharco et al., 2023a), PEM (Zhang et al., 2023a), and TIES-Merging (Yadav et al., 2024) enhance model fusion through parameter composition, thereby improving model adaptability. Model Evolver (Du et al., 2024c, 2023, 2024a) dynamically evolves model parameters, while Model Tailor (Zhu et al., 2024) mitigates catastrophic forgetting in multimodal tasks through model patching, decoration, and post-training. Tall-masks (Wang et al., 2024) offers efficient masking for model compression, and MATS (Tam et al., 2024) employs a conjugate gradient method to match task parameter subspaces.

In conclusion, our research focuses on leveraging pre-trained model parameters, as this approach provides better transfer performance and greater efficiency at a lower cost.

2.2 Model Pruning

Model pruning can be broadly classified into two main approaches. The first approach encompasses traditional model pruning techniques. This includes structured pruning methods such as SliceGPT (Ashkboos et al., 2024) and LLM-pruner (Ma et al., 2023), as well as unstructured pruning techniques like SparseGPT (Frantar and Alistarh, 2023), Wanda (Sun et al., 2023), GRAIN (Yang et al., 2023), GBLM-Pruner (Das et al., 2023), and OWL (Yin et al., 2023).

The second approach focuses on pruning fine-tuned models given a pretrained model. For instance, Model Grafting (Panigrahi et al., 2023) creates a mask to identify the most critical parameters for a specific task by optimizing the target task loss. TIES (Yadav et al., 2024) addresses interference issues that arise after magnitude pruning. DARE (Yu et al., 2023a) aligns task vector parameters with the expected model output by randomly selecting and rescales them. Model Tailor (Zhu et al., 2024) produces a sparse mask based on salience and sensitivity scores, while Talls Mask (Wang et al., 2024) combines the merged model with an additional mask to localize task information, effectively reducing storage costs.

In this paper, we propose a novel pruning approach that is simple, efficient, and robust by searching for weight coefficients within neural parameter subspaces.

3 Methodology

3.1 Problem Setting

Here, we consider knowledge transfer, fusion and compression of a set of tasks $\{T_1, \ldots, T_n\}$ and various pre-trained models like ViT (Dosovitskiy et al., 2021), T5 (Raffel et al., 2020), or Llama2 (Touvron et al., 2023). To begin, each pre-trained model is optimized on task-specific data, which can be performed either by fine-tuning the entire model or by using a parameter-efficient fine-tuning (PEFT) method (Liu et al., 2022b; Hu et al., 2022). During this process, the trainable parameters θ were initialized with θ_{pre} (the pre-trained state) and subsequently updated to θ_{ft} (the fine-tuned state).

Recent research introduced the concept of task vectors (Ilharco et al., 2023a), which has been applied in various knowledge transfer, fusion, and compression tasks. For a specific task T, the task vector $\tau \in \mathbb{R}^d$ is defined as the difference between the fine-tuned weights θ_i and the pre-trained weights θ_{pre}, i.e., $\tau = \theta - \theta_{\text{pre}}$. This captures the changes during the fine-tuning phase for each task-specific model. Building on this idea, a pruned fine-tuned model $\hat{\theta_{\text{ft}}}$ can be obtained by first deriving the pruned task vector $\hat{\tau}$, as defined in the equation below:

$$\hat{\theta_{\text{ft}}} = \theta_{\text{pre}} + \hat{\tau} \tag{1}$$

3.2 Neural Parameter Search for Pruning

Given that different parameter subspaces of task vectors contribute variably to fine-tuning performance, we first decomposed the task vector τ into M independent parameter subspaces q_m by ranking the parameters based on their magnitude and then dividing them according to these ranks, summarized as $\tau = \sum_{m=1}^{M} q_m$. Next, to enable more effective pruning, we reallocated weights for each subspace to obtain a new task vector:

$$\tau = \sum_{m=1}^{M} w_m * q_m \tag{2}$$

while $*$ denotes scalar multiplication of a vector element-wise.

Initially, all weight coefficients were initialized to 1, after which we used an evolutionary algorithm

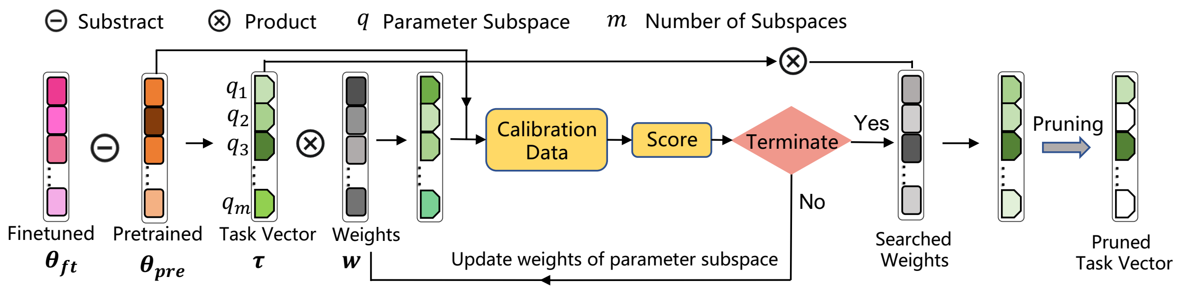

Figure 3: The framework of Neural Parameter Search enhances the efficiency of pruning fine-tuned models. This is achieved by searching and reweighting the neural parameters of task vectors within low-rank subspaces.

to search for a more optimal set of weight coefficients. The optimization process aims to find the best set $\{w_1, \ldots, w_m\}$, seeking optimal validation accuracy, and ultimately maximizing performance on calibration data with the adjusted fine-tuned model, as shown in Figure 3.

In most of our experiments, we employed Covariance Matrix Adaptive Evolution Strategies (CMA-ES) (Hansen and Ostermeier, 1996), a probabilistic, population-based optimization algorithm. CMA-ES dynamically adjusts the search distribution through a covariance matrix, updating the mean and covariance at each iteration to effectively exploit the structure of the search space for obtaining optimal candidate solutions. When the evolutionary algorithm has approximately converged, we combined the optimized weight coefficients with the task vector and the pre-trained model to obtain an adjusted model:

$$\theta_{\text{ft}} = \theta_{\text{pre}} + \sum_{m=1}^{M} w_m * q_m \tag{3}$$

Finally, we pruned the fine-tuned model based on the magnitude of its adjusted parameters after the search. We define the sparsity ratio as r, where $0 < r \leq 1$, and compute a mask m to select the most important neural parameters. This mask is derived using the following equation:

$$m_d = \begin{cases} 1, & \text{if } \tau_d \geq \text{sorted}(\tau)[r \times d] \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The final pruned fine-tuned model is then given by:

$$\hat{\theta_{\text{ft}}} = \theta_{\text{pre}} + m \odot \tau \tag{5}$$

while $\odot$ represents the Hadamard product.

This final model can subsequently be applied to scenarios such as knowledge transfer, fusion, and compression. To evaluate the pruning efficiency of the NPS method, we applied it to a pre-trained vision model, ViT-B/32, which was fine-tuned on various tasks. We then assessed the results of different pruning methods on the respective benchmarks for each task. The reported results are the average performance across eight fine-tuned models under varying levels of pruning sparsity, as illustrated in Figure 4. In comparison with baseline methods like TIES (Yadav et al., 2024) and DARE (Yu et al., 2023a), our findings indicated that when the pruning sparsity ratio exceeds 0.2, most methods maintain performance comparable to the fine-tuned models. However, as the sparsity ratio drops below 0.2, accuracy tends to decline rapidly. Notably, our NPS method demonstrates greater tolerance to lower sparsity ratios, preserving the original model's accuracy even at a sparsity ratio of 0.04.

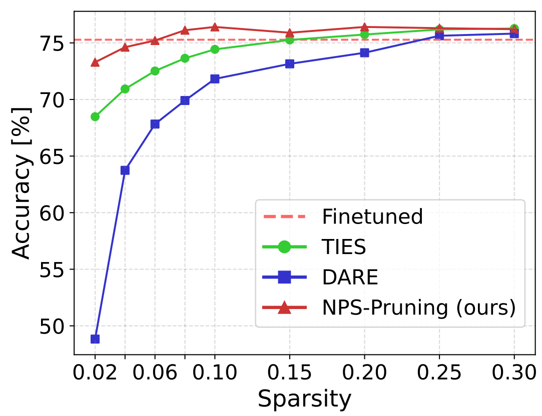

Figure 4: Performance variations of different methods with changes in sparsity ratio. Our NPS method exhibits higher tolerance to varying levels of sparsity.

3.3 Applications

Building on the significant improvement in pruning efficiency for fine-tuned models, we present three application scenarios for our proposed NPS method in the context of knowledge transfer, model fusion, and compression when pre-trained models and task-specific data for fine-tuning are available.

Knowledge Transfer. Fine-tuning language models on new, unseen data often leads to a decline in performance on the original tasks. Moreover, previous research (Zhu et al., 2024) indicates that fine-tuned models have low knowledge representation efficiency, containing a large number of redundant parameters that offer little benefit for new tasks. Removing these redundant parameters can minimize interference when integrating with the pre-trained model. Therefore, combining a pruned fine-tuned model with the original pre-trained model can enhance its resistance to catastrophic forgetting during knowledge transfer. We propose applying `NPS` to the parameters of the task vector before integrating them into the pre-trained model, as shown below:

$$\hat{\theta}_{\text{ft}} = \theta_{\text{pre}} + \lambda \cdot m \odot \tau \tag{6}$$

Here, λ is a hyperparameter used to rescale the neural parameters within the pruned task vector.

Knowledge Fusion. The knowledge fusion problem involves how to combine the finetuned model sets $\{\theta_1, \ldots, \theta_n\}$ to form a new model θ_m, without the need to retrain using the initial training data for each task, and ensuring that θ_m can simultaneously perform tasks $\{1, \ldots, n\}$. The multi-task model merging via task vectors is expressed as:

$$\theta_m = \theta_{\text{pre}} + \sum\nolimits_{i=1}^{n} (\lambda_i \cdot m_i \odot \tau_i) / \sum\nolimits_{i=1}^{n} \lambda_i \tag{7}$$

Here, λ_i is the coefficient for a specific pruned task vector, which can be optimized using evolutionary strategies to obtain an optimal set of $\{\lambda_1, \ldots, \lambda_n\}$ with the maximum validation accuracy for the final merged model.

Knowledge Compression. Pruning fine-tuned models is an effective strategy for compressing checkpoints. By applying sparsity masks to weights and storing only the masked values, we can preserve full performance while greatly reducing storage.

In term of storage for $\{\theta_t\}_{t=1}^{T}$, we only need to store the pre-trained model θ_{pre}, the task vectors τ, and the binary masks m for each task. For multi-task evaluation, fine-tuned models can be reconstructed by adding only the important subsets of task-specific vectors to the shared pretrained parametersθ_{pre}:

$$\hat{\theta}_{\text{ft}_1}, \ldots, \hat{\theta}_{\text{ft}_n} = \theta_{\text{pre}} + [m_1 \odot \tau_1, \ldots, m_n \odot \tau_n] \tag{8}$$

4 Experiment

4.1 Evaluation Settings

We expect that `NPS` will provide significant benefits for developers in three main areas: First, in experiments with multimodal large language models (MLLMs) using the LLaVA framework, our approach preserved performance even at a sparsity level of 10%. This highlights its effectiveness in mitigating catastrophic forgetting. Second, in knowledge fusion, it consistently outperforms existing merging techniques across various modalities, domains, and model sizes. Lastly, for knowledge compression, it achieves superior accuracy and storage efficiency compared to baselines on ViT-based vision tasks. More information on implementation details can be found in Appendix C.

4.2 Baseline Methods

Our baselines are categorized into three primary areas: knowledge transfer for mitigating catastrophic forgetting, knowledge fusion, and compression. For knowledge transfer, we compare our approach against Standard **Fine-tuning**, **Model Grafting** (Panigrahi et al., 2023), Drop & Rescale (**DARE**) (Yu et al., 2023a), and **Model Tailor** (Zhu et al., 2024). In the domain of knowledge fusion, we assess various methods such as **Simple Averaging** (Wortsman et al., 2022), **Fisher Merging** (Matena and Raffel, 2022), **RegMean** (Jin et al., 2023), **Task Arithmetic** (Ilharco et al., 2023a), **Ties-Merging** (Yadav et al., 2024), **PCB Merging** (Du et al., 2024b) and **Consensus Merging** (Wang et al., 2024). Notably, Task Arithmetic, Ties-Merging, Consensus Merging, and our proposed `NPS` are all based on task vectors, making them training-free and lightweight. For knowledge compression, we evaluate our method against several model merging techniques and their combinations with **Talls Mask** (Wang et al., 2024). Detailed information on these baselines can be found in Appendix D.

4.3 Results on Knowledge Transfer

Following (Liu et al., 2023a), we conduct knowledge transfer experiments using LLaVA-1.5 (Vicuna-7B). Both the projector and LLM parameters of the model are fine-tuned. The pre-trained datasets include VQAv2 (Goyal et al., 2017), GQA (Hudson and Manning, 2019), Vizwiz (Gurari et al., 2018), SQA (Lu et al., 2022), TextVQA (Singh et al., 2019), POPE (Li et al.,

Table 1: Average performance and H-score on LLaVA-1.5 (Vicuna-7B) with a sparsity ratio $r = 10\%$. “#Params" refers to the number of parameters modified. The optimal and sub-optimal results are denoted by boldface and underlining.

Method	#Params	Pre-trained tasks								Target task		
		VQAv2	GQA	VizWiz	SQA	TextVQA	POPE	MM-Bench	MM-Bench-CN	Flickr30k	Avg	Hscore
Zero-shot	-	78.52	61.97	50.0	70.17	58.28	85.97	64.78	58.51	18.62	42.33	29.05
Fine-tune	2.7B	68.61	49.01	27.24	63.86	40.03	79.73	59.02	50.17	77.1	56.42	63.40
DARE[ICML24]	273M	78.12	59.25	48.9	64.92	57.12	84.86	64.77	57.47	25.6	60.12	36.64
Grafting[ICML23]	273M	74.48	58.28	43.16	66.82	52.56	80.35	64.52	55.49	58.2	61.56	60.03
Model Tailor[ICML24]	273M	73.21	52.49	42.28	67.15	43.89	82.88	63.40	56.15	75.4	<u>61.87</u>	<u>66.94</u>
NPS (ours)	273M	74.3	52.52	43.1	66.12	43.93	83.23	64.52	57.51	76.2	**62.38**	**67.54**

Method	#Params	Pre-trained tasks								Target task		
		VQAv2	GQA	VizWiz	SQA	TextVQA	POPE	MM-Bench	MM-Bench-CN	OKVQA	Avg	Hscore
Zero-shot	-	78.52	61.97	50.0	70.17	58.28	85.97	64.78	58.51	0.14	27.94	33.09
Fine-tune	2.7B	69.1	48.61	30.35	41.03	42.13	72.33	32.79	43.47	46.27	47.34	46.87
DARE[ICML24]	273M	78.04	61.65	49.19	67.58	57.91	86.44	65.03	58.16	0.83	58.31	1.64
Grafting[ICML23]	273M	75.23	58.42	43.27	67.26	53.51	85.29	62.16	54.42	30.8	58.93	41.25
Model Tailor[ICML24]	273M	76.25	60.39	46.49	69.51	54.88	85.44	63.32	54.21	38.1	<u>60.95</u>	<u>47.71</u>
NPS (ours)	273M	76.81	60.94	48.1	71.32	56.34	87.23	64.77	57.5	38.4	**62.38**	**48.38**

2023b), MM-Bench (Liu et al., 2023b), and MM-Bench-CN (Zhang et al., 2023b). We then fine-tune LLaVA on Flickr30k (Young et al., 2014) and OKVQA (Marino et al., 2019) tasks, which are not included in the model's pre-training datasets. The performance of the fine-tuned model is evaluated on these and other datasets.

For evaluation, we use both the arithmetic and harmonic means (Zhu et al., 2024) of performance across pre-trained and target tasks, referred to as average performance and H-score. As shown in Table 1, our NPS method effectively mitigates catastrophic forgetting in MLLMs, outperforming current fine-tuning and forgetting mitigation techniques at a sparsity level of 10%. While further fine-tuning to improve performance on new tasks often deteriorates the model's effectiveness on pre-trained tasks, NPS successfully balances targeted optimization with the preservation of pre-trained performance. It achieves superior average metrics, improving by 1.5% and 1.4%, respectively, demonstrating its capability to enhance task-specific performance while maintaining robustness.

4.4 Results on Knowledge Fusion

To empirically validate the effectiveness of NPS, we conducted extensive experiments to compare it with existing model merging techniques. Our results highlight the advantages of our approach across both cross-task and cross-domain perspectives. Detailed information about the datasets used is provided in Appendix E.

Merging NLP Models. In the NLP domain, we follow the experimental setup outlined in (Yadav et al., 2024). We use the T5-base and T5-large models (Raffel et al., 2020), fine-tuning each on seven diverse tasks, including question answering, paraphrase identification, sentence completion, and coreference resolution. Table 2 demonstrates that merging fully fine-tuned T5-base and T5-large models using NPS results in an average performance improvement of 2.1% for T5-base and 1.6% for T5-large across the seven tasks.

Merging PEFT Model Adapters. Based on (Yadav et al., 2024), we explore parameter merging for efficient fine-tuning using the $(IA)^3$ method (Liu et al., 2022b), a type of Parameter-Efficient Fine-Tuning (PEFT) that extends base model activations with learned vectors. We use the T0-3B model (Sanh et al., 2022) and fine-tune $(IA)^3$ on training sets from eleven diverse datasets, including tasks such as sentence completion and natural language inference. For the $(IA)^3$ experiments, we report median scores across all templates for each dataset. As shown in Table 2, NPS improves average performance by 1.4% across 11 tasks compared to the top baseline.

Merging LLMs. In our experiment, we combined three specialized large language models built on the Llama-2-7b architecture (Touvron et al., 2023), each focusing on a different area: Chinese language proficiency[1], mathematical reasoning (Yu et al., 2023b)[2], and code generation (Rozière et al.,

[1] https://huggingface.co/LinkSoul/Chinese-Llama-2-7b

[2] https://huggingface.co/meta-math/MetaMath-7B-V1.0

Table 2: Comparison of different model merging methods across various fine-tuning configurations and modalities, with average performance reported for different tasks. The optimal and sub-optimal results are denoted by boldface and underlining.

Settings (→)	7 **NLP** Tasks		11 **PEFT** Tasks	3 **LLM** Tasks	8 **Vision** Tasks		5 **Emotion** Domains	
Method (↓)	T5-Base	T5-Large	$(IA)^3$	LLaMa2	ViT-B/32	ViT-L/14	T5-Base	RoBERTa-Base
Fine-tuned	83.1	88.9	71.4	40.4	90.5	94.2	51.38	49.38
Multitask	83.6	88.1	73.1	-	88.9	93.5	47.75	49.06
Averaging [ICML22]	65.3	54.7	57.9	30.3	65.8	79.6	23.2	38.3
Fisher Merging [NeurIPS22]	68.3	68.7	62.2	-	68.3	82.2	26.1	38.1
RegMean [ICLR23]	72.7	79.8	58.0	-	71.8	83.7	34.2	38.4
Task Arithmetic [ICLR23]	73.0	80.2	63.9	30.4	70.1	84.5	33.6	38.3
Ties-Merging [NeurIPS23]	73.6	80.3	66.8	34.2	73.6	86.0	34.5	39.7
Consensus TA [ICML24]	73.1	80.2	65.8	33.5	73.5	85.8	33.9	39.2
Consensus TIES [ICML24]	73.4	80.5	66.6	34.4	73.3	86.2	34.4	39.8
NPS (ours)	**75.7** (+2.1)	**82.1** (+1.6)	**68.2** (+1.4)	**35.3** (+0.9)	**76.5** (+3.0)	**87.6** (+1.4)	**35.7** (+1.3)	**40.9** (+0.9)

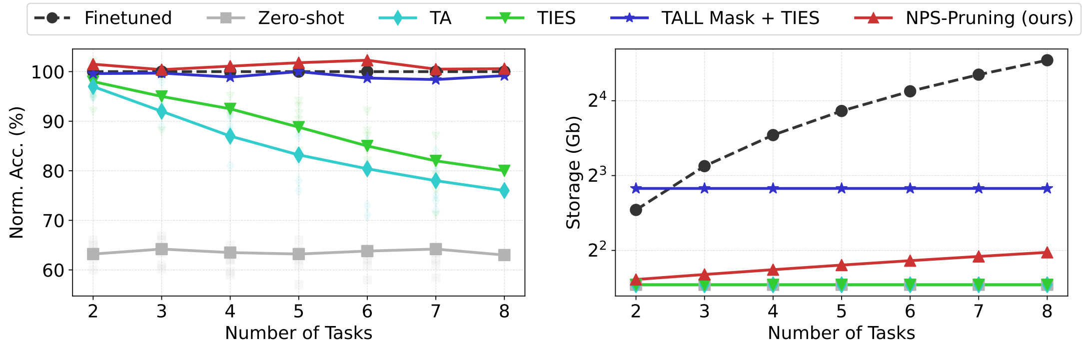

Figure 5: Averaged normalized accuracy and storage cost versus the number of tasks on computer vision benchmarks. Our proposed NPS method consistently preserves initial performance across various task combinations while significantly compressing the fine-tuned checkpoints.

2023)[3]. We assessed the performance of each model using specific benchmarks: CMMLU (Li et al., 2023a) for Chinese, GSM8K (Cobbe et al., 2021) for mathematics, and HumanEval (Chen et al., 2021) for code generation. As indicated in Table 2, our method NPS resulted in an average performance improvement of 0.9%.

Merging Vision Models. For image classification tasks, we adhered to the experimental setup outlined by (Ilharco et al., 2022, 2023a). We employed two versions of the CLIP model (Radford et al., 2021), specifically using ViT-B/32 and ViT-L/14 as visual encoders. The visual encoders were fine-tuned on eight tasks sourced from (Radford et al., 2021), while the text encoder remained unchanged. This approach covered a range of classification domains, such as remote sensing, traffic classification, and satellite imagery recognition. Our method achieved a 3.0% improvement over the top baseline on ViT-B/32 and a 1.4% improvement on ViT-L/14.

[3] https://huggingface.co/qualis2006/llama-2-7b-int4-python-code-18k

Merging Emotion Domains. We carried out further experiments to evaluate the effectiveness of various methods in merging five domain-specific emotion classification models. In line with the methodology of RegMean (Jin et al., 2023), we used the Roberta-base and T5-base models, along with five preprocessed datasets from (Oberländer and Klinger, 2018). Our analysis presents the average accuracy on in-domain datasets achieved by different model merging techniques. Additionally, we conducted experiments with multiple random seeds and reported the average results across five seeds. As detailed in Table 2, our approach surpasses the best baseline by 1.3% on Roberta-base and 0.9% on T5-base.

4.5 Results on Knowledge Compression

We conducted experiments using eight different ViT-B/32 models, each fine-tuned on distinct vision tasks, and tested the performance and compression efficiency across various numbers of tasks. For each task quantity, five random combinations were selected, and the average results were reported. As shown in Figure 8, both TALL-Mask and NPS

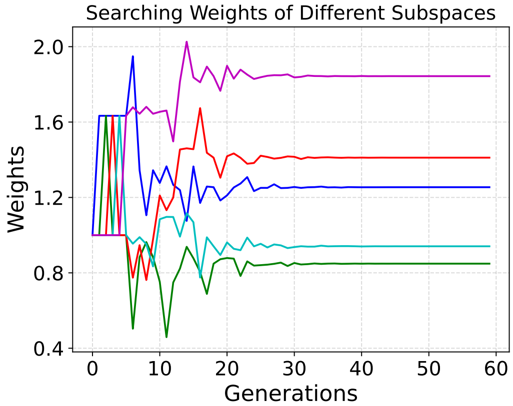

Figure 6: Searching for the weights of neural parameters across different task vector subspaces.

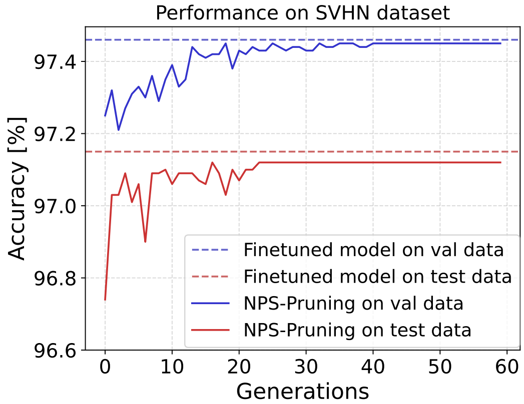

Figure 7: Performance convergence of the pruned fine-tuned model as the number of generations increases.

maintain around 99% normalized accuracy across all cases, with virtually no performance degradation as the number of tasks increases.

In terms of storage, our method significantly reduces costs compared to storing individual fine-tuned models, with the savings becoming more pronounced as the number of tasks increases. The TALL Mask + TIES method consistently consumes a high amount of storage, even when the number of tasks is small. In contrast, our approach requires storage that increases gradually with the number of tasks. While methods like Task Arithmetic have lower storage demands, they suffer from a noticeable drop in accuracy. Overall, our method achieves an optimal balance on the Pareto front, effectively retaining performance while minimizing total storage costs. More results about knowledge compression are provided in supplemental materials Appendix A.

5 Analysis

Search Visualization. To better understand the workflow of our method, we visualized the pruning process for a ViT-B/32 model fine-tuned on the SVHN dataset, setting the sparsity ratio to 0.1. We divided the task vector into five subspaces based on their magnitude values and then continuously updated the weights of these subspaces to explore higher validation scores. It can be observed that the weight values stabilize as the number of generations increases, as shown in Figure 6, and the pruned model's accuracy also gradually converges to a stable value, as shown in Figure 7.

Time complexity. The total time required for the overall `NPS` strategy is

$$T_{\text{total}} = \text{Generations} \times (T_{\text{pruning}} + T_{\text{validate}}) \quad (9)$$

where generations represents the number of generations needed for searching, which is a pre-set value and varies with different experiment settings. The pruning time mainly depends on the number of model parameters and the size of the model population, while the validation time primarily depends on the volume of inference data and the inference speed. We have organized ablation study and reported the number of generations and time required in our experiments, as shown in Appendix B.

Advantages. Our method offers several notable benefits, making it an efficient, flexible, and practical solution for various use cases.

- **Gradient-Free Operation**: `NPS` method operates without gradient calculations, making it lightweight and minimizing memory usage.
- **Practicality and Ease of Implementation**: The method is straightforward to implement and integrates easily into various applications.
- **Broader Applicability and Stable Performance**: Unlike theoretical pruning methods, our approach is more versatile and provides consistent results across various applications.

6 Conclusions

This study highlights the significance of pruning fine-tuned models when pretrained model is available. We introduce Neural Parameter Search (NPS) as an efficient technique for this task. Our approach facilitates multi-task model fusion, compression, and robust knowledge transfer by searching neural parameters within task vector subspaces. Experimental results demonstrate that `NPS` significantly enhances performance across various knowledge transfer scenarios.

Acknowledgements

This work was supported by National Science Foundation of China (62476070), Shenzhen Science and Technology Program (JCYJ20241202123503005, GXWD20231128103232001, ZDSYS20230626091203008, KQTD20240729102154066) and Department of Science and Technology of Guangdong (2024A1515011540).

Limitations

While our method offers advantages such as gradient-free operation, ease of implementation, and broad applicability, it also has certain limitations:

- **Dependence on Pretrained Models**: Our approach relies on pretrained models as a reference. If the fine-tuned model deviates significantly from the original, it may hinder effective knowledge transfer, fusion, and compression.
- **Validation Data Requirements**: The method requires additional validation data to guide the search process. The quality and quantity of this data directly impact pruning effectiveness and overall performance.
- **Computational Overhead of the Search Process**: Although the method is gradient-free, the search process introduces a time cost, which varies depending on task complexity. This trade-off should be considered when deploying the method in resource-constrained environments.

Ethical Considerations

Our research is based on publicly available and safe datasets and models. However, the applicability of `NPS` may be limited to similar datasets or domains. Its performance on other datasets remains uncertain, and applying it to privacy-sensitive or high-risk scenarios may pose risks. We recommend caution and thorough validation to ensure accuracy and reliability in such cases.

References

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 579–586.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024. Slicegpt: Compress large language models by deleting rows and columns. *Preprint*, arXiv:2401.15024.

Jinhe Bi, Yifan Wang, Danqi Yan, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. 2025a. Prism: Self-pruning intrinsic selection method for training-free multimodal data selection. *arXiv preprint arXiv:2502.12119*.

Jinhe Bi, Yujun Wang, Haokun Chen, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. 2024. Visual instruction tuning with 500x fewer parameters through modality linear representation-steering. *arXiv preprint arXiv:2412.12359*.

Jinhe Bi, Danqi Yan, Yifan Wang, Wenke Huang, Haokun Chen, Guancheng Wan, Mang Ye, Xun Xiao, Hinrich Schuetze, Volker Tresp, et al. 2025b. Cotkinetics: A theoretical modeling assessing lrm reasoning process. *arXiv preprint arXiv:2505.13408*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning (ICML)*, pages 1597–1607.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. 2017. Remote sensing image scene classification: Benchmark and state of the art. *In Proceedings of the IEEE*, pages 1865–1883.

Leshem Choshen, Elad Venezian, Noam Slonim, and Yoav Katz. 2022. Fusing finetuned models for better pretraining. *arXiv preprint arXiv:2204.03044*.

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

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

Rocktim Jyoti Das, Liqun Ma, and Zhiqiang Shen. 2023. Beyond size: How gradients shape pruning decisions in large language models. *Preprint*, arXiv:2311.04902.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

GuoDong Du, HaoJian Deng, JiaHao Su, and Yuan Huang. 2023. End-to-end rain streak removal with raw images. *arXiv preprint arXiv:2312.13304*.

Guodong Du, Runhua Jiang, Senqiao Yang, Haoyang Li, Wei Chen, Keren Li, Sim Kuan Goh, and Ho-Kin Tang. 2024a. Impacts of darwinian evolution on pre-trained deep neural networks. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1907–1912. IEEE.

Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim Kuan Goh, Ho-Kin Tang, Daojing He, and Min Zhang. 2024b. Parameter competition balancing for model merging. *Advances in Neural Information Processing Systems (NeurIPS)*, 37.

Guodong Du, Jing Li, Hanting Liu, Runhua Jiang, Shuyang Yu, Yifei Guo, Sim Kuan Goh, and Ho-Kin Tang. 2024c. Knowledge fusion by evolving weights of language models. *arXiv preprint arXiv:2406.12208*.

Zitao Fang, Guodong Du, Shuyang Yu, Yifei Guo, Yiwei Zhang, Jing Li, Ho-Kin Tang, and Sim Kuan Goh. 2025. Disentangling task interference within neurons: Model merging in alignment with neuronal mechanisms. *arXiv preprint arXiv:2503.05320*.

Ronald A Fisher. 1922. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character (PTRSL)*, pages 309–368.

Elias Frantar and Dan Alistarh. 2023. SparseGPT: Massive language models can be accurately pruned in one-shot. *arXiv preprint arXiv:2301.00774*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 6904–6913.

Hao Guo, Zihan Ma, Zhi Zeng, Minnan Luo, Weixin Zeng, Jiuyang Tang, and Xiang Zhao. 2025a. Each fake news is fake in its own way: An attribution multi-granularity benchmark for multimodal fake news detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 228–236.

Weiyang Guo, Jing Li, Wenya Wang, YU LI, Daojing He, Jun Yu, and Min Zhang. 2025b. Mtsa: Multi-turn safety alignment for llms through multi-round red-teaming. *arXiv preprint arXiv:2505.17147*.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3608–3617.

Nikolaus Hansen and Andreas Ostermeier. 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation (ICEC)*, pages 312–317.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing (STAEORS)*, pages 2217–2226.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning (ICML)*, pages 2790–2799.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 6700–6709.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023a. Editing models with task arithmetic. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023b. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. 2022. Patching open-vocabulary models by interpolating weights. *In Proceedings of Neural Information Processing Systems (NeurIPS)*, pages 29262–29277.

Runhua Jiang, Guodong Du, Shuyang Yu, Yifei Guo, Sim Kuan Goh, and Ho-Kin Tang. 2024. Cade: Cosine annealing differential evolution for spiking neural network. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2022. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 8082–8090.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 554–561.

Yann LeCun. 1998. The mnist database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.

Junlin Lee, Yequan Wang, Jing Li, and Min Zhang. 2024. Multimodal reasoning with multimodal knowledge graph. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the International Conference on the Principles of Knowledge Representation and Reasoning (KR)*.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023a. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*.

Junlin Li, Guodong DU, Jing Li, Sim Kuan Goh, Wenya Wang, Yequan Wang, Fangming Liu, Ho-Kin Tang, Saleh Alharbi, Daojing He, and Min Zhang. 2025. Multi-modality expansion and retention for llms through parameter merging and decoupling. *Preprint*, arXiv:2505.17110.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang. 2021. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, pages 370–403.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *In Proceedings of Neural Information Processing Systems (NeurIPS)*, pages 1950–1965.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022b. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *In Proceedings of Neural Information Processing Systems (NeurIPS)*, pages 1950–1965.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2023b. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *In Proceedings of Neural Information Processing Systems (NeurIPS)*, pages 2507–2521.

Yifan Lu, Yigeng Zhou, Jing Li, Yequan Wang, Xuebo Liu, Daojing He, Fangming Liu, and Min Zhang. 2025. Knowledge editing with dynamic knowledge graphs for multi-hop question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24741–24749.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *In Proceedings of Neural Information Processing Systems (NeurIPS)*, pages 21702–21720.

Xuetao Ma, Wenbin Jiang, and Hua Huang. 2025. Problem-solving logic guided curriculum in-context learning for llms complex reasoning. *arXiv preprint arXiv:2502.15401*.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition (CVPR)*, pages 3195–3204.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The CommitmentBank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung (SUB)*, pages 107–124.

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *In Proceedings of Neural Information Processing Systems (NeurIPS)*, pages 17703–17716.

Saif M. Mohammad. 2012. #emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (SEM)*, pages 246–255.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, page 7.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Laura Ana Maria Oberländer and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 2104–2119.

Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. 2023. Task-specific skill localization in fine-tuned language models. *arXiv preprint arXiv:2302.06600*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: The word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of International conference on machine learning (ICML)*, pages 8748–8763.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, pages 1–67.

Feiliang Ren, Longhui Zhang, Shujuan Yin, Xiaofeng Zhao, Shilei Liu, Bochao Li, and Yaduo Liu. 2021. A novel global feature-oriented relational triple extraction model based on table filling. *arXiv preprint arXiv:2109.06705*.

Feiliang Ren, Longhui Zhang, Xiaofeng Zhao, Shujuan Yin, Shilei Liu, and Bochao Li. 2022. A simple but effective bidirectional framework for relational triple extraction. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 824–832.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, pages 99–106.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Klaus R Scherer and Harald G Wallbott. 1994. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology (PSP)*, page 310.

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. 2018. Tackling the story ending biases in the story cloze test. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 752–757.

Zesheng Shi, Yucheng Zhou, and Jing Li. 2025. Safety alignment via constrained knowledge unlearning. *Preprint*, arXiv:2505.18588.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 8317–8326.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. 2011. The german traffic sign recognition benchmark: a multi-class classification competition. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. Quartz: An open-domain dataset of qualitative relationship questions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5940–5945.

Derek Tam, Mohit Bansal, and Colin Raffel. 2024. Merging by matching models in task parameter subspaces. *Transactions on Machine Learning Research (TMLR)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. 2024. Localizing task information for improved model merging and compression. *arXiv preprint arXiv:2405.07813*.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 23965–23998.

M. Xia, Z. Zhong, and D. Chen. 2022. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1513–1528.

Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. 2016. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision (IJCV)*, pages 3–22.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *In Proceedings of Neural Information Processing Systems (NeurIPS)*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2013–2018.

Ziqing Yang, Yiming Cui, Xin Yao, and Shijin Wang. 2023. Gradient-based intra-attention pruning on pretrained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Mykola Pechenizkiy, Yi Liang, Zhangyang Wang, and Shiwei Liu. 2023. Outlier weighed layerwise sparsity (owl): A missing secret sauce for pruning llms to high sparsity. In *arXiv preprint arXiv:2310.05175*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics (TACL)*, pages 67–78.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2023a. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *arXiv preprint arXiv:2311.03099*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023b. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhi Zeng, Minnan Luo, Xiangzheng Kong, Huan Liu, Hao Guo, Hao Yang, Zihan Ma, and Xiang Zhao. 2024. Mitigating world biases: A multimodal multi-view debiasing framework for fake news video detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6492–6500.

Jinghan Zhang, Junteng Liu, Junxian He, et al. 2023a. Composing parameter-efficient modules with arithmetic operation. *In Proceedings of in Neural Information Processing Systems (NeurIPS)*, pages 12589–12610.

Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Hang Yan, et al.

2023b. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Lei Zhao, Junlin Li, Lianli Gao, Yunbo Rao, Jingkuan Song, and Heng Tao Shen. 2022. Heterogeneous knowledge network for visual dialog. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(2):861–871.

Changshi Zhou, Rong Jiang, Feng Luan, Shaoqiang Meng, Zhipeng Wang, Yanchao Dong, Yanmin Zhou, and Bin He. 2025a. Dual-arm robotic fabric manipulation with quasi-static and dynamic primitives for rapid garment flattening. *IEEE/ASME Transactions on Mechatronics*, pages 1–11.

Changshi Zhou, Haichuan Xu, Jiarui Hu, Feng Luan, Zhipeng Wang, Yanchao Dong, Yanmin Zhou, and Bin He. 2025b. Ssfold: Learning to fold arbitrary crumpled cloth using graph dynamics from human demonstration. *IEEE Transactions on Automation Science and Engineering*, 22:14448–14460.

Didi Zhu, Zhongyisun Sun, Zexi Li, Tao Shen, Ke Yan, Shouhong Ding, Chao Wu, and Kun Kuang. 2024. Model tailor: Mitigating catastrophic forgetting in multi-modal large language models. In *Proceedings of Forty-first International Conference on Machine Learning (ICML)*.

Appendix

This paper enhances the pruning efficiency of fine-tuned models through **N**eural **P**arameter **S**earch and applies this approach to various scenarios, including knowledge transfer, fusion, and compression, with the assistance of pre-trained models. The appendix is organized based on the following contributions:

- Appendix A (Additional Results) provides additional experimental results on knowledge compression as well as task-level results from the knowledge fusion experiments.
- Appendix B (Additional Analysis) includes ablation studies, hyperparameter analysis, and time cost evaluation for the search process.
- Appendix C (Implementation Details) outlines the computational resources and runtimes, along with the training details and evaluation metrics.
- Appendix D (Baselines) provides a detailed baseline description.
- Appendix E (Datasets) provides a detailed dataset description.

A Additional Results

A.1 Additional Results on Compression

In our NLP experiments, particularly in the knowledge compression scenarios involving large language models, we present additional results, as shown in Appendix Tables 3. These results demonstrate that our method maintains the performance of the previous best compression approach, TALLS Mask+TIES, while significantly reducing storage consumption.

A.2 Comprehensive Task-Level Results

We present task-level results for all knowledge fusion experiments in Section 4.4. Detailed task-level outcomes for T5-Base, T5-Large (Raffel et al., 2020), IA3 (Liu et al., 2022b), ViT-B/32, and ViT-L/14 (Dosovitskiy et al., 2021) are provided in Appendix Tables 4, 5, 6, 7, and 8, respectively. We also provide radar charts to compare the results of merging vision tasks, as illustrated in Appendix Figure 8. While previous baseline methods exhibit inconsistent performance and struggle with certain tasks, our method proves to be more robust, delivering near-optimal results across all tasks.

Table 3: Comparison of different knowledge compression methods across various modalities, with average performance reported for different tasks. The optimal results are denoted by boldface. Please refer to Section 4.5 for more details.

Settings (→)	7 **NLP** Tasks				3 **LLM** Tasks		8 **Vision** Tasks			
	T5-Base		T5-Large		LLaMa2		ViT-B/32		ViT-L/14	
Method (↓)	Acc.(%)↑	Bits(Gb)↓	Acc.(%)↑	Bits(Gb)↓	Acc.(%)↑	Bits(Gb)↓	Acc.(%)↑	Bits(Gb)↓	Acc.(%)↑	Bits(Gb)↓
Fine-tuned	83.1	47.8	88.9	169.1	40.4	629.6	90.5	23.3	94.2	79.1
Zero-shot	$53.5_{(64.4)}$	7.1	$53.1_{(59.7)}$	25.1	$15.3_{(37.9)}$	215.6	$62.3_{(68.8)}$	3.6	$74.5_{(79.1)}$	11.0
Task Arithmetic[ICLR23]	$73.0_{(87.8)}$	7.1	$80.2_{(90.2)}$	25.1	$30.4_{(75.2)}$	215.6	$70.1_{(77.5)}$	3.6	$84.5_{(89.7)}$	11.0
TIES[NeurIPS23]	$73.6_{(88.6)}$	7.1	$80.3_{(90.3)}$	25.1	$34.2_{(84.7)}$	215.6	$73.6_{(81.3)}$	3.6	$86.0_{(91.3)}$	11.0
Talls+TIES[ICML24]	$82.6_{(99.4)}$	15.2	$88.3_{(99.3)}$	54.3	$39.5_{(97.8)}$	442.3	$90.2_{(99.7)}$	7.1	$93.6_{(99.4)}$	23.1
NPS (ours)	$\mathbf{82.9}_{(\mathbf{99.8})}$	11.1	$\mathbf{88.8}_{(\mathbf{99.9})}$	39.2	$\mathbf{40.5}_{(\mathbf{100.2})}$	276.3	$\mathbf{90.9}_{(\mathbf{100.4})}$	5.9	$\mathbf{94.3}_{(\mathbf{100.1})}$	18.0

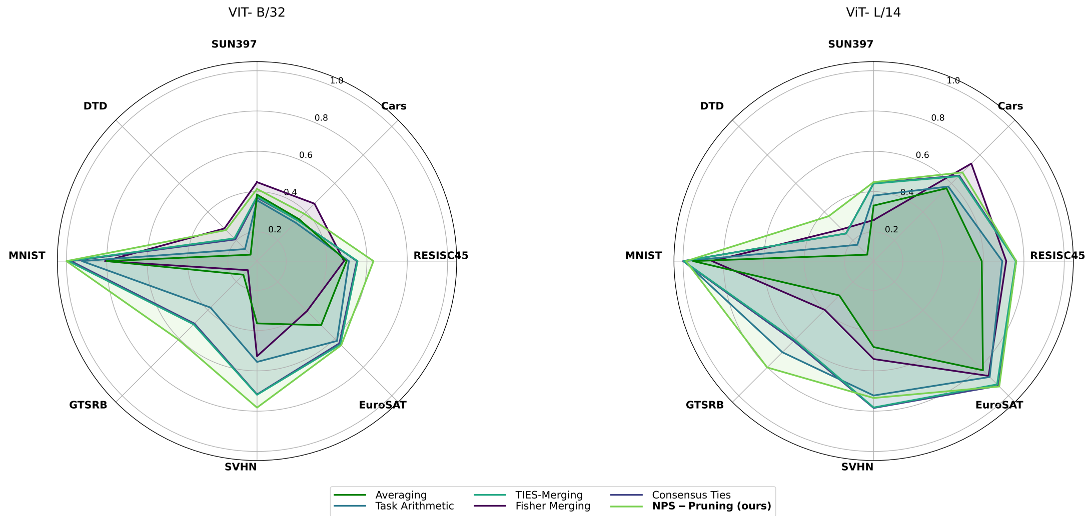

Figure 8: Test set performance when merging ViT-B/32 and ViT-L/14 models on eight image classification tasks.

Table 4: Test set performance when merging T5-base models on seven NLP tasks. Please refer to Section 4.4 for more details.

Task(→) Method(↓)	Average	Test Set Performance						
		paws	qasc	quartz	story_cloze	wiki_qa	winogrande	wsc
Zeroshot	53.5	49.9	35.8	53.3	48.1	76.2	50	61.1
Fine-tuned	83.1	94.6	98.4	81.1	84.9	95.8	64.5	62.5
Multitask	83.6	94	97.9	82.5	86.7	95	64.1	65.3
Averaging[ICML22]	65.3	67.4	83.4	60.8	50.3	93.2	51.7	50.0
Fisher Merging[NeurIPS22]	68.3	66.7	85.6	63.5	57.1	90.1	54.2	60.8
RegMean[ICLR23]	72.7	77.2	**93.8**	63.6	64.6	90.4	58.4	60.7
Task Arithmetic[ICLR23]	73.0	69.6	91.5	**67.3**	76.1	91.3	58.3	56.9
Ties-Merging[NeurIPS23]	73.6	82.2	84.8	66.1	73.5	87.0	60.2	61.1
Consensus Ties[NeurIPS23]	73.4	**82.3**	84.5	65.7	73.4	86.8	**60.3**	60.5
NPS (ours)	**75.6**	79.1	93.3	65.9	**76.2**	**89.9**	59.9	**63.9**

B Additional Analysis

B.1 Ablation Studies

Our method incorporates several key factors, including the number of subspaces, the volume of the calibration dataset, and the sparsity of pruning levels. We conducted ablation studies on these elements, with the results presented in Appendix Table 9, 10, 11. Specifically, we tested our approach on knowledge fusion across eight ViT models for vision tasks.

B.2 Hyperparameters

Due to the hyperparameter sensitivity in task vector-based model merging methods, we provide the optimal values of λ and r as determined by our experiments, as outlined in Tab. 12. For Task Arithmetic, we explored λ within the range of 0.2 to 1.5, using

Table 5: Test set performance when merging T5-large models on seven NLP tasks. Please refer to Section 4.4 for more details.

Task(→) Method(↓)	Average	Test Set Performance paws	qasc	quartz	story_cloze	wiki_qa	winogrande	wsc
Zeroshot	53.1	58.2	54.2	54.1	54.3	70.9	49.2	63.9
Fine-tuned	88.9	94.5	98.3	88.5	91.4	96.2	74.5	79.2
Multitask	88.1	94.2	98.5	89.3	92	95.4	73.5	73.6
Averaging[ICML22]	54.7	57.2	26.4	71.4	54.8	86.6	50.2	36.1
Fisher Merging[NeurIPS22]	68.7	68.4	83	65.5	62.4	94.1	58.2	49.2
RegMean[ICLR23]	79.8	**83.9**	97.2	73.2	82.6	94.1	63.2	64.4
Task Arithmetic[ICLR23]	80.2	77.6	96.6	**75.1**	85.6	93.8	61.8	70.8
Ties-Merging[NeurIPS23]	80.3	78.2	97.5	72.8	83.7	**94.5**	64.5	70.8
Consensus Ties[NeurIPS23]	80.5	78.4	97.7	72.6	83.7	**94.8**	64.6	71.2
NPS (ours)	**82.1**	82.1	**98.4**	72.3	**85.7**	94.1	**67.2**	**75.0**

Table 6: Test set performance when merging (IA)^3 models on eleven tasks. Please refer to Section 4.4 for experimental details.

Task(→)	Average	Natural Language Inference					Sentence Completion			Co-reference		WSD
Method(↓)		RTE	CB	ANLI1	ANLI2	ANLI3	COPA	Hella.	Story.	WSC	Wino.	WiC
Zeroshot	53.1	58.2	54.2	35.5	34.4	34.4	75.0	39.2	86.5	63.9	51.2	51.9
Fine-Tuned	71.4	82.7	95.8	70.4	46.5	53.0	85.3	44.4	95.0	65.3	75.1	71.7
Averaging[ICML22]	57.9	81.2	58.3	43.3	39.1	40.0	80.9	40.1	92.4	52.8	53.8	55.0
Fisher Merging[NeurIPS22]	62.2	83.3	83.3	45.9	41.0	42.2	83.1	42.2	94.1	58.3	56.7	54.2
RegMean[ICLR23]	58	81.2	58.3	43.3	39.2	40.2	80.9	40.1	92.5	53.5	53.8	55
Task Arithmetic[ICLR23]	63.9	74.1	83.3	60.8	49.4	50.0	87.5	41.5	**95.3**	49.3	62.8	49.1
Ties-Merging[NeurIPS23]	66.8	78.6	**87.5**	66.6	**51.3**	**51.5**	81.7	43.2	90.9	57.6	67.0	58.4
Consensus Ties[ICML24]	66.6	78.5	87.3	66.4	51.1	51.2	81.6	**43.4**	90.2	57.3	67.1	58.3
NPS (ours)	**68.2**	80.1	83.5	**67.3**	51.2	49.8	**88.4**	42.6	92.8	61.9	**67.5**	**64.8**

Table 7: Test set performance when merging ViT-B/32 models on 8 vision tasks. Please refer to Section 4.4 for more details.

Task(→) Method(↓)	Average	Test Set Performance SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD
Individual	90.5	75.3	77.7	96.1	99.7	97.5	98.7	99.7	79.4
Multitask	88.9	74.4	77.9	98.2	98.9	99.5	93.9	72.9	95.8
Averaging[ICML22]	65.8	65.3	63.4	71.4	71.7	64.2	52.8	87.5	50.1
Fisher Merging[NeurIPS22]	68.3	**68.6**	**69.2**	70.7	66.4	72.9	51.1	87.9	**59.9**
RegMean[ICLR23]	71.8	65.3	63.5	75.6	78.6	78.1	67.4	93.7	52
Task Arithmetic[ICLR23]	70.1	63.8	62.1	72	77.6	74.4	65.1	94	52.2
Ties-Merging[NeurIPS23]	73.6	64.8	62.9	74.3	78.9	83.1	71.4	97.6	56.2
Consensus Ties[NeurIPS23]	73.3	64.5	63.0	74.1	78.5	83.0	71.1	96.9	55.8
NPS (ours)	**76.5**	66.8	65.4	**78.5**	**79.2**	**86.5**	**77.1**	**98.1**	59.3

Table 8: Test set performance when merging ViT-L/14 models on 8 vision tasks. Please refer to Section 4.4 for more details.

Task(→) Method(↓)	Average	Test Set Performance SUN397	Cars	RESISC45	EuroSAT	SVHN	GTSRB	MNIST	DTD
Fine-tuned	94.2	82.3	92.4	97.4	100	98.1	99.2	99.7	84.1
Multitask	93.5	90.6	84.4	99.2	99.1	99.6	96.3	80.8	97.6
Averaging[ICML22]	79.6	72.1	81.6	82.6	91.9	78.2	70.7	97.1	62.8
Fisher Merging[NeurIPS22]	82.2	69.2	**88.6**	87.5	93.5	80.6	74.8	93.3	70
RegMean[ICLR23]	83.7	73.3	81.8	86.1	97	88	84.2	98.5	60.8
Task Arithmetic[ICLR23]	84.5	74.1	82.1	86.7	93.8	87.9	86.8	98.9	65.6
Ties-Merging[NeurIPS23]	86	76.5	85	89.4	95.9	90.3	83.3	99	68.8
Consensus Ties[NeurIPS23]	86.2	76.6	85.2	**89.5**	96.3	**90.4**	83.6	**99.1**	68.8
NPS (ours)	**87.6**	**76.8**	86.1	**89.5**	**96.5**	88.4	**91.1**	98.5	**73.7**

Table 9: The performance of NPS in knowledge fusion on vision tasks across varying volumes of calibration datasets.

Volume	Ties-Merging	1/4	1/2	1
ViT-B/32	73.6	75.9	76.3	76.5
ViT-L/14	86.0	87.1	87.5	87.6

Table 10: The performance of NPS in knowledge fusion on vision tasks across varying numbers of subspaces.

Numbers	Ties-Merging	1	2	4	8
ViT-B/32	73.6	74.8	75.6	76.2	76.5
ViT-L/14	86.0	86.9	87.3	87.5	87.6

a step size of 0.1. In the cases of TIES-Merging and NPS, we varied the mask ratios r across $\{0.05, 0.1, 0.2\}$, while λ was searched within the range of 0.8 to 2.5 with a step size of 0.1. For knowledge compression using NPS, we fixed the ratio r at 0.05 to minimize storage costs.

B.3 Time cost

The total time required for the overall NPS strategy is

$$T_{\text{total}} = \text{Generations} \times (T_{\text{pruning}} + T_{\text{validate}}) \quad (10)$$

where generations represents the number of generations needed for searching, which is a pre-set value and varies with different experiment settings. The pruning time mainly depends on the number of model parameters and the size of the model population, while the validation time primarily depends on the volume of inference data and the inference speed. We have organized and reported the number of generations and the time required for each task in Appendix Table 13. As shown, our method typically requires only a few hours (2-6 hours) to complete, even for large language models.

B.4 More Related Work

In recent years, significant progress has been made in knowledge transfer, model fusion, and compression techniques (Li et al., 2025), enabling the efficient deployment of large-scale models (Shi et al., 2025). Several studies have shown that fine-tuning pre-trained models with a small amount of data can significantly improve their zero-shot generalization ability (Ren et al., 2022, 2021). Parameter-efficient tuning methods, such as LoRA and Adapter, have demonstrated strong performance in low-resource scenarios (Bi et al., 2025b,a, 2024). Moreover,

Table 11: The performance of NPS in knowledge fusion on vision tasks with different sparsity pruning ratios r.

Ratios	0.03	0.05	0.1	0.2	0.3
ViT-B/32	75.8	76.5	76.3	75.2	72.1
ViT-L/14	86.9	87.6	87.2	86.5	83.4

contrastive learning has been widely employed in unsupervised settings to facilitate model compression and fusion (Guo et al., 2025b,a; Ma et al., 2025). In the area of model fusion, Fisher-weighted averaging estimates parameter importance using information-theoretic measures, while RegMean offers a closed-form solution based on local regression. Other works, including task arithmetic, PEM, and TIES-Merging, enhance generalization by linearly combining parameters across tasks (Zhao et al., 2022; Lee et al., 2024; Lu et al., 2025). Model Evolver dynamically optimizes fusion trajectories, and Model Tailor mitigates catastrophic forgetting in multimodal tasks through patching, decoration, and post-training strategies (Zeng et al., 2024; Zhou et al., 2025a,b).

C Implementation details

C.1 Computational Resources and Runtimes

Our experiments were conducted on Nvidia A6000 GPUs with 48GB of RAM. Depending on the dataset size, fine-tuning the T5-Base and T5-Large models for single tasks took between 15 minutes and 2 hours, while fine-tuning the multitask checkpoint took around eight hours. The fine-tuned $(IA)^3$ models were provided by Yadav et al. (2024).[4]. We also used vision models ViT-B/32 and ViT-L/14 as provided by Ilharco et al. (2023a).[5]. Merge experiments were highly efficient, with evaluations for RoBerta-base, T5-Base, T5-Large, ViT-B/32, and ViT-L/14 models taking less than 2 minutes. However, two specific experiments required more time: (1) Evaluating $(IA)^3$ models took about one hour for 11 datasets due to the need to use multiple templates from prompt sources and compute median results across them. (2) Validation on LLMs (LLaMa2) was also slow, usually requiring about 40 minutes for evaluating 3 datasets.

[4] https://github.com/prateeky2806/ties-merging
[5] https://github.com/mlfoundations/task_vectors#checkpoints

Table 12: λ and pruning ratio r for NPS

Task (→) Method (↓)	7 **NLP** Tasks T5-Base	T5-Large	11 **PEFT** Tasks $(IA)^3$	3 **LLM** Tasks LLaMa2	8 **Vision** Tasks ViT-B/32	ViT-L/14
Task Arithmetic[ICLR23] [λ]	0.4	0.5	0.5	0.3	0.3	0.3
Ties-Merging[NeurIPS23] [λ, r]	[1.7, 0.1]	[2.4, 0.05]	[1.7, 0.1]	[1.0, 0.1]	[1.0, 0.1]	[1.1, 0.05]
NPS for fusion (ours) [λ, r]	[1.9, 0.05]	[2.2, 0.05]	[1.8, 0.1]	[0.9, 0.1]	[1.2, 0.05]	[1.2, 0.05]
NPS for compression (ours) [r]	0.05	0.05	-	0.05	0.05	0.05

Table 13: Time Costs for NPS.

Task (→) Method (↓)	7 **NLP** Tasks T5-Base	T5-Large	11 **PEFT** Tasks $(IA)^3$	3 **LLM** Tasks LLaMa2	8 **Vision** Tasks ViT-B/32	ViT-L/14
Time for Pruning	5 secs	9 secs	1 secs	113 secs	4 secs	7 secs
Time for Validation	4 mins	7 mins	15 mins	12 mins	6 mins	9 mins
Generations	30	50	20	20	30	30
Total Time for NPS	126 mins	358 mins	300 mins	278 mins	183 mins	273 mins

C.2 Training details

We trained the T5-base and T5-large models for up to 75,000 steps, using a batch size of 1024 and a learning rate of 0.0001. Early stopping with a patience of 5 was employed to prevent overfitting. Training was conducted in bfloat16 to conserve GPU memory, with a sequence length capped at 128 tokens. For the PEFT configuration of the $(IA)^3$ approach on the T0-3B model, the batch size was set to 16 for training and 32 for evaluation, while maintaining a learning rate of 0.0001. The early stopping patience was extended to 10 due to the model's complexity. We didn't use any learning rate scheduler or weight decay during training. For large language models, we used fine-tuned checkpoints from Huggingface[6].

In the cross-domain merging experiments, we fine-tuned the RoBERTa-base model with an initial learning rate of 1e-5 and the T5-base model at 1e-4, using the AdamW optimizer. The learning rate was gradually increased during the first 6% of training steps, then linearly decreased to zero. Both models were trained with a batch size of 16 over 30 epochs for emotion classification, with performance evaluated at the end of each epoch, resuming from the best checkpoint.

C.3 Evaluation Metrics

Normalized Accuracy. We report both normalized and absolute accuracies. Normalization is based on of the individual fine-tuned models.

$$\text{Acc.} = \frac{1}{N}\sum_{n=1}^{N} \frac{\underset{x\sim\mu_n}{\text{acc}}\left[f_{\text{merged}}(x)\right]}{\underset{x\sim\mu_n}{\text{acc}}\left[f_{\text{fine-tuned}}(x)\right]} \tag{11}$$

[6]https://huggingface.co/

H-Score. To rigorously evaluate our method's ability to mitigate catastrophic forgetting in MLLMs, we use two key metrics: Average Performance and the H-score (Zhu et al., 2024). The H-score, a novel metric, provides a balanced assessment by calculating the harmonic mean between the average performance on original tasks, $\text{Avg}(P_{\text{origin}})$, and on target tasks, $\text{Avg}(P_{\text{target}})$. The formula for the H-score is as follows:

$$P_H = \frac{2\times\text{Avg}(P_{\text{origin}})\times\text{Avg}(P_{target})}{\text{Avg}(P_{\text{origin}})+\text{Avg}(P_{\text{target}})}. \tag{12}$$

The H-score was introduced to avoid overemphasizing the performance of original tasks, especially as their number grows.

Storage Cost. This section show the calculation of the storage cost for each method in Section 4.5 and Appendix A Tab. 3. Let N be the number of tasks, P be the number of all parameters, P' be the number of trainable parameters in the model, and F be the number of frozen parameters in the model. Assuming one float parameter takes 32 bits, for each method, their respective storage cost for T tasks is calculated as:

- Fine-tuned models: $32(NP' + F)$. $32NP'$ is for storing T trainable parameters and $32F$ is for storing frozen parameters.
- Task arithmetic: $32P$; Stores a single model.
- Ties-merging: $32P$; Stores a single model.
- Consensus Ties: $32P$; Stores a single model.
- Zero-shot: $32P$; Stores a single model.

- TALL Mask + Ties: $(64+N)P' + 32F$; $64P' + 32F$is for storing zeroshot model and multi-task vector, while NP' is for storing T binary masks.

- NPS: $32P + (r*32+1)NP'$; r is the sparsity pruning ratio.

D Baseline details

We provied a detailed baseline description. Our experiments encompass seven comparison methods:

- **Individual** means that each task uses an independent fine-tuned model, which has no interference between tasks, but cannot perform multiple tasks simultaneously.
- **Traditional MTL** collects the original training data of all tasks together to train a multi-task model. It can be used as a reference *upper bound* for model merging work.
- **Weight Averaging** is the simplest method of model merging, which directly averages the parameters of multiple models using $\theta_m = \sum_{t=1}^{n} \theta_t / n$, calculating the element-wise mean of all individual models. It can be used as a *lower bound* for model merging. (Choshen et al., 2022; Wortsman et al., 2022).
- **Fisher Merging** (Matena and Raffel, 2022) calculates the Fisher information matrix (Fisher, 1922) $\hat{F}_t = \mathbb{E}_{x\sim D_t}\mathbb{E}_{y\sim p_{\theta_t}(y|x)} \nabla_{\theta_t} (\log p_{\theta_t}(y|x_t))^2$ to measure the importance of each parameter when merging models for task t, where and model merging is performed according to the guidance of this importance.
- **RegMean** (Jin et al., 2023) imposes a constraint when merging models, that is, the L_2 distance between the merged model's and the individual models' activations. It computes a least-squares solution as $\theta_m = (\sum_{t=1}^{n} X_t^T X_t)^{-1} \sum_{t=1}^{n} (X_t^T X_t \theta_t)$, where X_t is the input activation of the corresponding layer.
- **Task Arithmetic** (Ilharco et al., 2023a) first defines the concept of "task vectors" and merges these vectors into a pre-trained model to execute multi-task learning. The model is produced by scaling and adding the task vectors to the initial model as $\theta_m = \theta_{\text{init}} + \lambda * \sum_{t=1}^{n} \tau_t$.
- **Ties-Merging** (Yadav et al., 2024) further solves the task conflict problem in Task Arithmetic (Ilharco et al., 2023a). It eliminates redundant parameters and resolves symbol conflicts through three steps: Trim, Elect Sign, and Disjoint Merge.
- **AdaMerging** automatically learns a merging coefficient for each layer of each task vector in Task Arithmetic (Ilharco et al., 2023a).
- **LoraHub** (Huang et al., 2023) employs Low-rank Adaptations to dynamically combine task-specific modules for cross-task generalization, and adapts to new tasks by configuring $\theta' = \sum_{k=1}^{K} w_k \cdot \theta_k$.
- **DARE** (Yu et al., 2023a) sets the majority of delta parameters to zero and rescale the rest by $\theta' = \theta \cdot (1/(1-p))$ where p is the proportion of delta parameters dropped, therefore efficiently reduces parameter redundancy.

E Datesets details

This section provides a detailed dataset description for our experiments.

NLP Tasks. Following TIES-Merging (Yadav et al., 2024), we choose seven datasets for merging NLP models: question answering (QASC (Khot et al., 2020), WikiQA (Yang et al., 2015), and QuaRTz (Tafjord et al., 2019)), paraphrase identification (PAWS (Zhang et al., 2019)), sentence completion (Story Cloze (Sharma et al., 2018)), and coreference resolution (Winogrande (Sakaguchi et al., 2021) and WSC (Levesque et al., 2012)).

PEFT Models. Following TIES-Merging (Yadav et al., 2024), we use eleven datasets including sentence completion (COPA (Roemmele et al., 2011), H-SWAG (Zellers et al., 2019), and Story Cloze (Sharma et al., 2018) datasets), natural language inference (ANLI (Nie et al., 2020), CB (Marneffe et al., 2019), and RTE (Giampiccolo et al., 2007)), coreference resolution (WSC (Levesque et al., 2012) and Winogrande (Sakaguchi et al., 2021)), and word sense disambiguation (WiC (Pilehvar and Camacho-Collados, 2019)).

Vision Tasks. Following Task Arithmetic (Ilharco et al., 2023a), we study multi-task model merging on eight image classification datasets below. Stanford Cars (Krause et al., 2013) is a car classification dataset consisting of 196 classes of cars. DTD (Cimpoi et al., 2014) is a texture classification dataset comprising 47 classes. EuroSAT (Helber et al., 2019) comprises 10 classes of geo-referenced satellite images. GTSRB (Stallkamp et al., 2011) includes 43 classes of traffic signs.

MNIST (LeCun, 1998) features grayscale images of handwritten digits across 10 classes. RESISC45 (Cheng et al., 2017) encompasses 45 classes of remote sensing image scenes. SUN397 (Xiao et al., 2016) consists of 397 classes of scene images. Lastly, SVHN (Netzer et al., 2011) encompasses 10 classes of real-world digital classification images.

Table 14: Statistics of emotion classification datasets.

	Train	Dev	Test
In-domain			
DialyDialog	72,085	10,298	20,596
CrowdFlower	27,818	3,974	7,948
TEC	14,735	2,105	4,211
Tales-Emotion	10,339	1,477	2,955
ISEAR	5,366	766	1,534

Emotion Classification. In order to investigate the performance of the sentiment classification task, following RegMean (Jin et al., 2023), we selected a diverse and challenging set of datasets. Among them, DailyDialogs (Li et al., 2017), CrowdFlower, TEC (Mohammad, 2012), Tales-Emotion (Alm et al., 2005), and ISEAR (Scherer and Wallbott, 1994) is utilized to train domain-specific model. For evaluation, we focus exclusively on the fundamental emotions: anger, disgust, fear, joy, sadness, and surprise. A detailed overview of the datasets and statistics is provided in Tab. 14.

LLMs.

- CMMLU (Li et al., 2023a) is a comprehensive Chinese evaluation benchmark specifically designed to assess language models' knowledge and reasoning abilities in a Chinese context. It covers 67 topics ranging from basic subjects to advanced professional levels.
- GSM8K (Cobbe et al., 2021) is a collection of 8.5K high-quality, linguistically varied math word problems from grade school, crafted by skilled human authors. The solutions predominantly require executing a series of basic arithmetic operations ($+$, $-$, $\times$, $\div$) to derive the final answer.
- HumanEval (Chen et al., 2021) is a dataset for evaluating code generation ability, containing 164 manually crafted programming problems covering aspects such as language understanding, reasoning, algorithms, and simple mathematics.