Dynamic Fisher-weighted Model Merging via Bayesian Optimization

Sanwoo Lee[1*] **, Jiahao Liu**[2]**, Qifan Wang**[3]**, Jingang Wang**[2]**, Xunliang Cai**[2]**, Yunfang Wu**[1†]
[1]School of Computer Science, Peking University; [2]Meituan; [3]Meta AI
{sanwoo, wuyf}@pku.edu.cn, wqfcr@fb.com
{liujiahao12,wangjingang02,caixunliang}@meituan.com

Abstract

The fine-tuning of pre-trained language models has resulted in the widespread availability of task-specific models. Model merging offers an efficient way to create multi-task models by combining these fine-tuned models at the parameter level, without the need for training data or joint training on multiple datasets. Existing merging approaches typically involve scaling the parameters model-wise or integrating parameter importance parameter-wise. Both approaches exhibit their own weaknesses, leading to a notable performance gap compared to multi-task fine-tuning. In this paper, we unify these seemingly distinct strategies into a more general merging framework, and introduce **D**ynamic **F**isher-weighted **M**erging (**DF-Merge**)[1]. Specifically, candidate models are associated with a set of coefficients that linearly scale their fine-tuned parameters. Bayesian optimization is applied to dynamically adjust these coefficients, aiming to maximize overall performance on validation sets. Each iteration of this process integrates parameter importance based on the Fisher information conditioned by the coefficients. Experimental results show that DF-Merge outperforms strong baselines across models of different sizes and a variety of tasks. Our analysis shows that the effectiveness of DF-Merge arises from the unified view of merging and that near-optimal performance is achievable in a few iterations, even with minimal validation data.

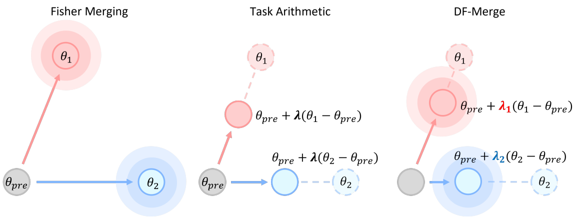
Fisher Merging
Task Arithmetic
DF-Merge
θ_1
θ_{pre}
θ_2
θ_1
$\theta_{pre} + \boldsymbol{\lambda}(\theta_1 - \theta_{pre})$
$\theta_{pre} + \boldsymbol{\lambda}(\theta_2 - \theta_{pre})$
θ_{pre}
θ_2
θ_1
$\theta_{pre} + \boldsymbol{\lambda_1}(\theta_1 - \theta_{pre})$
$\theta_{pre} + \boldsymbol{\lambda_2}(\theta_2 - \theta_{pre})$
θ_2

Figure 1: Comparison of DF-Merge with primary approaches in model merging. **left**: leverages parameter importance evaluated at the fine-tuned models. **middle**: uniformly scales fine-tuned models to alleviate parameter interference. **right**: DF-Merge optimizes distinct scaling coefficients and incorporates parameter importance evaluated at the scaled models.

1 Introduction

Modern transformer-based pre-trained language models (PLMs) (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020) have driven a paradigm shift towards fine-tuning PLMs for specific tasks, achieving state-of-the-art performance across various applications. The general-purpose representations learned through pretraining have significantly enhanced numerous downstream tasks, leading to the widespread development of fine-tuned expert models (Min et al., 2023). For example, over a million models have been uploaded to the Hugging Face repository (Wolf, 2019), with many publicly available for research study[2].

Most off-the-shelf models are fine-tuned independently for individual tasks, which limits their performance outside of their specialized domains. Ideally, models should be capable of handling multiple tasks relevant to a particular use case. Although multi-task learning (MTL) (Søgaard and Goldberg, 2016; Deng et al., 2019) offers a straightforward solution, it must require simultaneous access to the labeled datasets of all tasks and training over those datasets. This challenge is pronounced given the increasing difficulty of fine-tuning PLMs with their ever-growing sizes. More importantly, since the original training data for each model is often proprietary, it is time-consuming or even infeasible for users to label a large amount of data for MTL.

[*] work done during an internship at Meituan.
[†] Corresponding author.
[1] Code is available at https://github.com/sanwooo/df-merge

[2] https://huggingface.co/models

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 4923–4935
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

Model merging offers a cost-effective alternative for building multi-task models by combining off-the-shelf models in the parameter space without additional training. For example, this can be done by simply weight-averaging the *task vectors* (i.e. fine-tuned part of the parameters from the pre-trained model) (Ilharco et al., 2023; Yang et al., 2024). The success of model merging is supported by recent findings that the local minima, optimized from pre-trained parameters, are linearly connected in a flat basin of the loss landscape with no barriers in between (Neyshabur et al., 2020; Zhou et al., 2024). As a result, linearly interpolating between fine-tuned models potentially produces a well-behaved model with multi-task capabilities.

Despite the advantages of model merging, current methods still lag behind the performance of multi-task fine-tuned models. This shortfall can be attributed to the fact that existing approaches improve only specific aspects of model merging. These model merging methods can be divided into two groups: (1) scaling the task vectors *model-wise* (Ilharco et al., 2023; Yang et al., 2024; Liu et al., 2024), and (2) accounting for parameter importance *parameter-wise* (Matena and Raffel, 2022; Jin et al., 2023; Tam et al., 2024). We present a general merging framework that these two seemingly distinct approaches can be unified into. Building on this framework, we introduce **D**ynamic **F**isher-weighted **M**erging (**DF-Merge**) which leverages the strengths of both strategies, as illustrated in Figure 1. In essence, DF-Merge uses Bayesian optimization to adjust the scaling coefficients in order to maximize the overall performance, with each iteration targeting a low-loss basin informed by (approximated) Fisher information.

Experimental results show that DF-Merge significantly outperforms competitive baselines across PLMs of different sizes on a variety of tasks. Ablation study confirms that the components of DF-Merge collectively contributes to the performance, validating the advantage of the general merging framework. Additionally, our analysis demonstrates that DF-Merge can achieve near-optimal performance within just a few iterations using minimal validation data. Our contributions are summarized as follows:

- We formulate the two primary model merging approaches into an unified objective, achieving a more flexible and effective model merging framework.
- We introduce Bayesian optimization in model merging to identify the optimal coefficients which allows for direct maximization of non-differentiable metrics.
- Our DF-Merge approach achieves significant improvements over the baselines, making it an effective and efficient alternative over multi-task learning.

2 Model Merging Revisit

Notation. Let $net(\theta)$ be a neural network parameterized by $\theta \in \mathbb{R}^d$. Consider T task-specific models $\{net(\theta_i)\}_{i=1}^{T}$, each initialized from the same pre-trained model $net(\theta_{pre})$ and fine-tuned on the i-th task dataset $\mathcal{D}_i = \{x_i^{(j)}, y_i^{(j)}\}_{j=1}^{N_i}$ where N_i is the dataset's cardinality. The goal of model merging is to create a multi-task model $net(\theta^*)$ that is proficient in all tasks.

Task Arithmetic (TA). Ilharco et al. (2023) coined a concept of *task vector* which represents the direction in the parameter space that enhances a pre-trained model's performance on the task. In particular, the task vector τ_i for the i-th task is specified by the fine-tuned part of parameters from the pre-trained model, expressed as $\tau_i = \theta_i - \theta_{pre}$. Task vectors can be combined by arithmetic operations to steer the pre-trained model's behavior on various tasks. This concept has been extended to *model-wise* model merging (Yadav et al., 2024; Yu et al., 2024; Yang et al., 2024), in which multiple task vectors are added to the pre-trained parameters

$$\theta_{new} = \theta_{pre} + \lambda \sum_{i=1}^{M} \Phi(\tau_i) \tag{1}$$

where $\lambda \in \mathbb{R}$ is a scaling coefficient and Φ denotes additional operations on the task vectors such as trimming, electing (Yadav et al., 2024), or dropout (Yu et al., 2024). These operations are designed to reduce parameter interference across the fine-tuned models. While the original implementation optimizes for a single coefficient λ on held-out validations sets, this can be generalized to multiple coefficients $\theta_{new} = \theta_{pre} + \sum_{i=1}^{M} \lambda_i \Phi(\tau_i)$, which we name as **General Task Arithmetic** (**GTA**).

Fisher Information. Understanding the landscape of the loss function allows for better alignment of different models' parameters. The local curvature of a loss function $\ell(\theta)$ at the point θ is captured by its second-order derivatives $\nabla^2 \ell(\theta)$, denoted as the Hessian matrix $H_\theta \in \mathbb{R}^{d\times d}$. Then the expectation of Hessian over the data distribution $p_\theta(x, y)$ describes how sensitive the loss function is to the parameters in the data distribution modeled by θ, where highly sensitive parameters potentially imply greater importance.

Assuming the model is fine-tuned using negative log-likelihood loss $\ell(\theta) = -\log p(y|x, \theta)$, the expectation of H_θ can be efficiently computed by the Fisher information (FI):

$$F_\theta = \mathop{\mathbb{E}}_{x\sim q(x)} \left[\mathop{\mathbb{E}}_{y\sim p_\theta(y|x)} \nabla_\theta \ell(\theta) \nabla_\theta \ell(\theta)^\top \right] \quad (2)$$

which only requires computing the first-order derivatives.

As estimating the expectation over the input distribution $x \sim q(x)$ is intractable, F_θ is approximated with the empirical Fisher information $\hat{F}_\theta$

$$\hat{F}_\theta = \frac{1}{N} \sum_{i=1}^{N} \left[\mathop{\mathbb{E}}_{y\sim p_\theta(y|x^{(i)})} \nabla_\theta \ell(\theta) \nabla_\theta \ell(\theta)^\top \right] \quad (3)$$

Note that the expectation over y is not calculated over the true labels, but rather measured on the predictive distribution $y \sim p_\theta(y|x)$ parameterized by θ. In practice, $y \sim p_\theta(y|x)$ can either be modeled exactly or through sampling depending on the size of label space (Matena and Raffel, 2022).

Fisher Merging from Geometric Perspective. Tam et al. (2024) studied a geometric analysis of Fisher Merging (Matena and Raffel, 2022), by representing Fisher Merging (without some approximations) as

$$\theta^* = \left(\sum_{i=1}^{\mathrm{M}} Q_i \Lambda_i Q_i^\top \right)^{-1} \left(\sum_{i=1}^{\mathrm{M}} Q_i \Lambda_i Q_i^\top \theta_i \right) \quad (4)$$

where $Q_i \Lambda_i Q_i^\top$ is the eigendecomposition of F_{θ_i}. Inspecting this form, $Q_i \Lambda_i Q_i^\top$ upweights the "important" eigenvector component of θ_i, such that useful parameters are preserved during merging.

Building upon this insight, we consider the following geometric objective $g(\theta)$ and show that Fisher Merging is a natural result of minimizing it:

$$\theta^* = \arg\min_\theta \sum_{i=1}^{\mathrm{M}} \|\Lambda_i^{1/2}(Q_i^\top \theta_i - Q_i^\top \theta)\|^2 \quad (5)$$

which restricts θ to move along the loss-insensitive principal directions in the parameter space, as indicated by the eigenvectors associated with smaller eigenvalues. Given that each fine-tuned model θ_i represents a local minimum for its respective task, moving along loss-insensitive directions is helpful for preventing θ from increasing loss of each task, thereby balancing the fine-tuned models and potentially targeting a low-loss basin shared by all tasks.

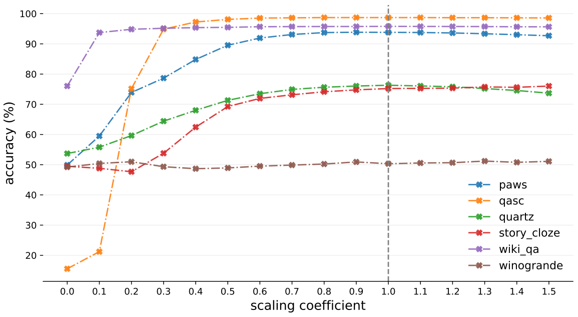

Figure 2: Accuracy (y-axis) of linear inter/extrapolation between the pre-trained model and fine-tuned models with varying coefficients λ (x-axis), T5-base.

As $g(\theta)$ is convex, setting its gradient to zero leads to the closed-form solution:

$$\begin{aligned} \frac{\partial g(\theta)}{\partial \theta} &= 2 \sum_{i=1}^{\mathrm{M}} \left[\Lambda_i^{1/2} Q_i^\top (\theta - \theta_i) \right]^\top \frac{\partial \Lambda_i^{1/2} Q_i^\top (\theta - \theta_i)}{\partial \theta} \\ &= 2 \sum_{i=1}^{\mathrm{M}} \left[\Lambda_i^{1/2} Q_i^\top (\theta - \theta_i) \right]^\top \Lambda_i^{1/2} Q_i^\top \\ &= 2 \sum_{i=1}^{\mathrm{M}} (\theta - \theta_i)^\top F_{\theta_i} = 0 \end{aligned}$$

which becomes equivalent to Fisher Merging:

$$\theta^* = \left(\sum_{i=1}^{\mathrm{M}} F_{\theta_i} \right)^{-1} \left(\sum_{i=1}^{\mathrm{M}} F_{\theta_i} \theta_i \right) \quad (6)$$

In practice, F_{θ_i} is replaced by its diagonal approximation to reduce computational complexity (Matena and Raffel, 2022), which can be seen as assuming independence between the parameters (i.e., $Q_i = I$) (Tam et al., 2024).

3 Dynamic Fisher-weighted Merging

Performance drop after merging models fine-tuned on different tasks occurs due to parameter interference, in which each task vector may represent a loss-increasing direction for the other tasks. Our pilot study (Figure 2) shows that linearly interpolating between a pre-trained model and a fine-tuned model reveals numerous alternative local minima, motivating us to search for a set of coefficients such that applying Fisher Merging on the models

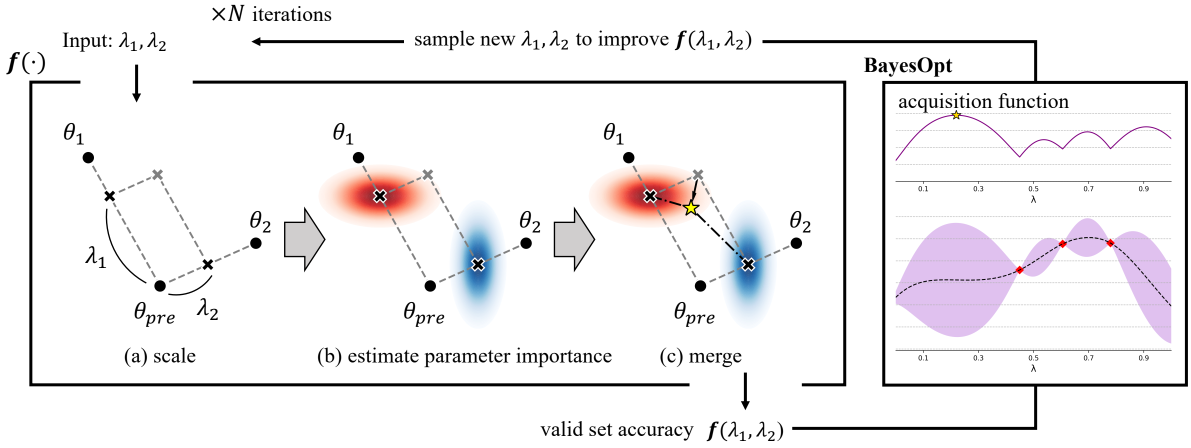
×N iterations
Input: λ_1, λ_2
sample new λ_1, λ_2 to improve $\boldsymbol{f}(\lambda_1, \lambda_2)$
$\boldsymbol{f}(\cdot)$
BayesOpt
acquisition function
0.1
0.3
0.5
0.7
0.9
λ
θ_1
θ_2
θ_{pre}
λ_1
λ_2
(a) scale
(b) estimate parameter importance
(c) merge
valid set accuracy $\boldsymbol{f}(\lambda_1, \lambda_2)$

Figure 3: An illustration of **DF-Merge**. (1) The black-box function $f(\cdot)$ takes coefficients as inputs to (a) scale the task vectors, and (c) merges models (yellow star) after (b) accounting for the parameter importance using FI, where a contour depicts the local loss landscape of a specific task. (2) The validation set accuracy $f(\cdot)$ is used by Bayesian optimization to suggest the best guess on the coefficients for the next iteration that improve $f(\cdot)$.

interpolated by the coefficients minimizes parameter interference. We apply Bayesian optimization for an efficient search of optimal coefficients. An overview of our proposed method, **DF-Merge**, is illustrated in Figure 3. In the following paragraphs, we present an unified view of model merging, then proceed to the details of DF-Merge, divided into merge function and coefficient optimization.

An Unified View of Model Merging. We show that Fisher Merging (parameter-wise) and Task Arithmetic (model-wise) both falls under the restricted cases of a more generic form of model merging. This generalized perspective offers a natural way to link both approaches. In particular, we propose a general function of model merging $f(\lambda_1, ..., \lambda_M; \theta_1, ..., \theta_M)$

$$f = \left(\sum_{i}^{M} C_{\theta_i}\right)^{-1} \left(M \sum_{i}^{M} C_{\theta_i} \cdot \lambda_i \tau_i\right) + \theta_{pre} \quad (7)$$

where C_{θ_i} is a covariance matrix (e.g., Fisher Information) that depends on θ_i.

This formulation recovers Averaging by setting $\lambda_i = 1/M$ and $C_{\theta_i} = I$. General Task Arithmetic (GTA) follows from $C_{\theta_i} = I$ while Fisher Merging is obtained with $C_{\theta_i} = \text{diag}(\hat{F}_{\theta_i})$ and $\lambda_i = 1/M$. GTA and Fisher Merging make orthogonal improvements over Averaging: GTA removes the implicit restriction of $\lambda_i = 1/M$, whereas Fisher Merging refines parameter importance by replacing $C_{\theta_i} = I$ with $\text{diag}(\hat{F}_{\theta_i})$.

Drawing from this insight, we propose the merge function of **DF-Merge** via linking the benefits of the two:

$$f = \left(\sum_{i}^{M} \text{diag}(\hat{F}_{\theta_i(\lambda_i)})\right)^{-1} \left(M \sum_{i}^{M} \text{diag}(\hat{F}_{\theta_i(\lambda_i)}) \lambda_i \tau_i\right) + \theta_{pre} \quad (8)$$

where $\text{diag}(\hat{F}_{\theta_i(\lambda_i)})$ is the diagonal Fisher Information estimated at $\theta_i(\lambda_i) := \lambda_i \tau_i + \theta_{pre}$. Intuitively, this allows Fisher Information to be estimated with varying λ_i along the path connecting θ_{pre} and θ_i, unlike Fisher Merging with fixed $\text{diag}(\hat{F}_{\theta_i(1)})$.

Coefficient Optimization. We employ Bayesian optimization to determine the coefficients $\{\lambda_i\}_{i=1}^{M}$ of Eq.8 that maximize average accuracy on the held-out validation sets. Unlike gradient descent, Bayesian optimization is well-suited for optimizing non-differentiable metrics like accuracy, precision or recall, which directly aligns with the goal of model merging. In addition, Bayesian optimization finds near-optimal coefficients within a few iterations, making it far more scalable than grid search as the number of models increases.

We utilize Gaussian Process to maximize the black box function $f_b(\lambda)$ ($\lambda := [\lambda_1, ..., \lambda_M] \in \mathbb{R}^M$) that returns a scalar metric (i.e., average accuracy) given the merging coefficients λ. Specifically, the Gaussian process prior is placed over the initial random observations on t points (Williams and Rasmussen, 2006; Frazier, 2018):

$$f_b(\lambda^{1:t}) \sim \mathcal{N}\left(\mu_0(\lambda^{1:t}), \Sigma_0(\lambda^{1:t}, \lambda^{1:t})\right) \quad (9)$$

where $\lambda^{1:t}$ is a compact representation of the collection of t points $[\lambda^1, ..., \lambda^t]$, and μ_0 and Σ_0 are the mean function and covariance function. Then, the *posterior distribution* of the value of the next point $f_b(\lambda^{t+1})$ is updated by the Bayes' Rule:

$$f_b(\lambda^{t+1}) | f_b(\lambda^{1:t}) \sim \mathcal{N}\left(\mu_t(\lambda^{t+1}), \sigma_t^2(\lambda^{t+1})\right) \tag{10}$$

where $\mu_t(\lambda^{t+1})$ and $\sigma_t^2(\lambda^{t+1})$ are defined as:

$$\begin{aligned}\mu_t(\lambda^{t+1}) &= \Sigma_0(\lambda^{t+1}, \lambda^{1:t})\Sigma_0(\lambda^{1:t}, \lambda^{1:t})^{-1} \\ &\quad \cdot (f_b(\lambda^{1:t}) - \mu_0(\lambda^{1:t})) + \mu_0(\lambda^{t+1}) \\ \sigma_t^2(\lambda^{t+1}) &= \Sigma_0(\lambda^{t+1}, \lambda^{t+1}) - \Sigma_0(\lambda^{t+1}, \lambda^{1:t}) \\ &\quad \cdot \Sigma_0(\lambda^{1:t}, \lambda^{1:t})^{-1}\Sigma_0(\lambda^{1:t}, \lambda^{t+1}).\end{aligned}$$

Subsequently, the next point λ^{t+1} to sample is determined by the *acquisition functions*, and we consider Expected Improvement (EI) (Frazier, 2018) and Upper Confidence Bound (UCB) (Srinivas et al., 2010) in our experiments. EI chooses λ^{t+1} such that it maximizes the expected value of improvement than the current best value $f_b^*(t)$ over its posterior distribution:

$$\arg\max_{\lambda^{t+1}} E_{f_b(\lambda^{t+1})}\left[\max(f_b(\lambda^{t+1}) - f_b^*(t), 0)\right]. \tag{11}$$

UCB selects λ^{t+1} such that it maximizes the peak of the confidence interval at λ^{t+1}:

$$\arg\max_{\lambda^{t+1}} \mu_t(\lambda^{t+1}) + \beta^{1/2}\sigma_t(\lambda^{t+1}) \tag{12}$$

where β is a constant that balances the exploration-exploitation tradeoff. The sampling process is repeated until it reaches the pre-defined number of iterations or the metric converges.

4 Experiments

4.1 Experimental Setup

Models and Datasets. We use **T5-base** and **T5-large** (Raffel et al., 2020) which are based on the encoder-decoder architecture and pre-trained on a large-scale corpus with denoising objectives. Both task-specific and multi-task models are fine-tuned on six datasets: PAWS (Zhang et al., 2019), QASC (Khot et al., 2020), QuaRTz (Tafjord et al., 2019), Story Cloze (Sharma et al., 2018), WikiQA (Yang et al., 2015) and Winogrande (Sakaguchi et al., 2021). These datasets cover a range of NLP tasks, including question answering, paraphrase identification, sentence completion, and coreference resolution. See Table 1 for the dataset statistics[3]. The inputs and outputs are formatted in natural language using the templates in PromptSource (Bach et al., 2022) toolkit. See details of training and testing in Appendix A. Note that training is only for simulating model merging under controlled environment, and we posit no access to the training data during merging.

Dataset	# train	# validation	# test	Task Type
PAWS	49401	8000	8000	Paraphrase Identification
QASC	8134	463	463	Question Answering
QuaRTz	2696	384	784	Question Answering
Story Cloze	1871	935	936	Sentence Completion
WikiQA	20360	2733	6165	Question Answering
Winogrande	40398	633	634	Coreference Resolution

Table 1: Dataset Statistics.

Evaluation Metric. All tasks are evaluated by accuracy.

Baselines. We compare our approach with several state-of-the-art baselines, including **Averaging** (Wortsman et al., 2022), **Fisher Merging** (Matena and Raffel, 2022), **Task Arithmetic** (Ilharco et al., 2023), **DARE** (Yu et al., 2024) and **TIES-Merging** (Yadav et al., 2024). TIES-Merging employs trim, elect, and disjoint mean operation to resolve parameter interference between the fine-tuned models. DARE randomly drops and re-scales the task vectors to sparsify them, potentially alleviating the parameter interference.

Implementation Details. **DF-Merge:** We optimize the coefficients using Bayesian Optimization package (Nogueira, 2014–) to maximize average accuracy on held-out validation sets. DF-Merge runs for 50 iterations, preceded by 10 random initialization steps, with coefficients constrained to the range of $[0, 1]$. For each iteration, diag($\hat{F}$) is computed exactly over the model's predictive distribution using 30 unlabeled validation samples. **Baselines:** The best coefficients of Task Arithmetic and TIES-Merging are determined by a grid search (TA: $[0, 1]$, TIES: $[0.8, 1.8]$) on validation sets with a step size of 0.1. DARE is applied on TA, with additional grid search over the drop rate p in $[0.1, 0.9]$ with a step size of 0.2. Unless otherwise stated, experimental results are averaged over five random runs with significance testing.

[3] For datasets without a publicly available labeled test set, the validation set is split into two halves to create new validation and test sets. For datasets with only validation and test sets, the validation set is used for training, and the test set is split into two halves to form new validation and test sets.

Model	Method	Valid. Set	PAWS	QASC	QuaRTz	Story Cloze	WikiQA	Winogrande	Avg.
T5-base	Zero-shot	-	49.89	15.55	53.70	49.47	76.04	49.21	48.98
	Fine-tune	-	93.81	98.70	76.30	75.21	95.79	50.32	81.69
	Multi-task	-	93.26	98.49	66.38	80.73	95.46	56.09	81.73
	Averaging	✗	68.92	82.33	59.72	49.74	94.30	50.79	67.63*
	Fisher Merging	✓	88.47	84.02	64.64	52.76	94.79	**51.04**	72.62*
	Task Arithmetic	✓	79.75	88.21	62.81	67.09	95.17	48.64	73.61*
	DARE	✓	79.80	87.82	62.86	67.35	95.16	48.96	73.66*
	TIES-Merging	✓	**90.18**	78.32	61.53	57.78	95.10	49.43	72.06*
	DF-Merge (EI)	✓	89.62	**97.37**	**68.21**	**68.97**	95.02	49.62	**78.14**
	DF-Merge (UCB)	✓	89.58	96.11	66.63	68.70	**95.19**	49.21	77.57
T5-large	Zero-shot	-	55.39	11.23	54.97	50.32	70.79	48.42	48.52
	Fine-tune	-	94.36	98.32	86.43	90.77	96.16	54.42	86.74
	Multi-task	-	94.29	99.18	83.11	89.83	95.94	67.54	88.31
	Averaging	✗	75.27	35.08	70.64	57.22	86.83	50.00	62.51*
	Fisher Merging	✓	67.70	61.64	81.30	68.35	89.16	51.10	69.88*
	Task Arithmetic	✓	90.86	95.46	73.05	84.78	93.24	**53.79**	81.86
	DARE	✓	91.01	95.68	72.07	84.66	93.20	52.84	81.58*
	TIES-Merging	✓	**93.12**	93.05	69.21	79.91	92.51	53.60	80.23*
	DF-Merge (EI)	✓	89.43	96.46	81.30	**86.99**	**95.15**	52.24	**83.59**
	DF-Merge (UCB)	✓	89.94	**96.76**	**81.68**	85.79	94.70	51.86	83.46

Table 2: Evaluation result (%) of DF-Merge and the baselines on six tasks. The best accuracy is bolded and the second-best accuracy is underlined, for each column of a type of model. *: Both DF-Merge (EI) and DF-Merge (UCB) significantly outperform the baseline ($p < 0.05$).

4.2 Main Results

Table 2 shows the performance of DF-Merge and the baselines. There are several key observations from the results. **First,** DF-Merge outperforms the baselines in average accuracy by large margins and the improvements are significant for almost all baselines. In particular, DF-Merge improves over the best baseline in the average accuracy by 4.48 point for T5-base and 1.73 point for T5-large. **Second,** DF-Merge narrows the gap with the oracle multi-task learning model by a substantial degree. For example, the gap in average accuracy can be narrowed down to 3.55 point for T5-base and 3.15 point for T5-large. This result indicates that DF-Merge can be a useful training-free alternative to multi-task learning in settings where a slight performance drop is permissible. **Third,** DF-Merge strikes an adequate balance between the performances across multiple tasks. For instance, the maximum drop in accuracy compared to the fine-tuned model among the six tasks is 8.09 point for T5-base and 4.75 point for T5-large, which are smaller than all baselines. In contrast, the baselines tend to build a multi-task model that excels in one task yet at the cost of compromising the performance of other tasks. For instance, Fisher Merging achieves a notable accuracy on QuaRTz (T5-large) while being much worse on the remaining tasks than the other methods.

Method	T5-base	T5-large
DF-Merge (EI)	78.14	83.59
w/o Fisher Information	76.80* (-1.34)	82.52 (-1.07)
w/o Bayesian Optimization	72.62* (-5.52)	69.88* (-13.71)
DF-Merge (UCB)	77.57	83.46
w/o Fisher Information	76.72* (-0.85)	82.49 (-0.97)
w/o Bayesian Optimization	72.62* (-4.90)	69.88* (-13.58)
Averaging	67.63	62.51

Table 3: Ablation of DF-Merge components, evaluated by the average test set accuracy (%). *: significant drop in performance after ablation ($p < 0.05$).

5 Analysis and Discussion

5.1 Ablation Study

We conduct an ablation study of DF-Merge components to understand each component's contribution to the final performance, as shown in Table 3. Removing Fisher Information from DF-Merge—i.e., using GTA and selecting coefficients via Bayesian optimization—results in a consistent drop in performance across different model sizes and acquisition functions. Notably, the performance drop is significant with EI, highlighting the importance of leveraging useful information from local loss curvature. Besides, removing Bayesian optimization from DF-Merge—i.e., Fisher Merging—causes significant drops in performance, indicating that DF-Merge benefits largely from a flexible coefficient search. To summarize, both Fisher information and Bayesian optimization are essential to the optimal

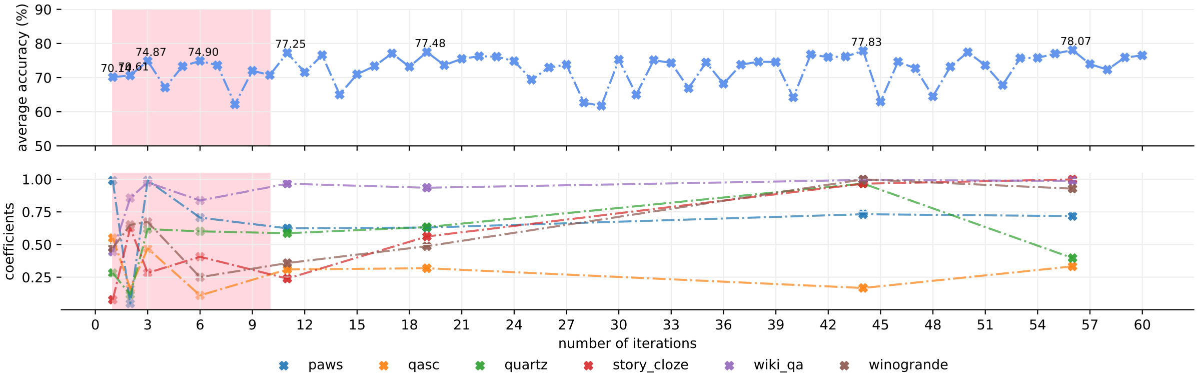

Figure 4: Bayesian optimization trajectory of DF-Merge (T5-base, UCB). Coefficients (**bottom**) and the average validation set accuracy (**top**) are rendered as a function of iterations. The coefficients with the new highest average accuracy up to their corresponding iteration are shown. The red area denotes initial random evaluations.

performance of DF-Merge.

5.2 Efficiency Analysis

Though DF-Merge effectively identifies optimal task vector coefficients in a vast search space, it still requires a number of merge-then-evaluate rounds with labeled validation sets, posing a challenge in terms of both computational budget and data labeling cost. Hence we examine whether the effectiveness of DF-Merge remains solid within a few number of iterations as well as with validations sets of reduced sizes. Results in this section are based on a single run with a fixed random seed.

Effect of the number of iterations. Figure 4 demonstrates that DF-Merge quickly achieves the near-optimal performance within a few number of iterations. In particular, after the initial evaluations on 10 random points (red area), it takes 9 iterations for DF-Merge to exploit previous observations and discover near-optimal coefficients with 0.59%p gap compared to the best ones (56th iteration).

The remaining iterations are responsible for a marginal improvement, indicating that the major enhancement in performance occurs in the early stage of optimization. We observe similar trends consistently when using EI as the acquisition function or using T5-large, shown in Appendix B. Hence the optimization may be terminated early to save much of the runtime and computational resources.

Effect of the validation set size. We randomly sample varying ratios of data from the validation set of each task with larger sampled sets containing the smaller ones, and observe the *test* set performance of DF-merge, as shown in Figure 5. Similar to the findings for the number of iterations, DF-merge efficiently achieves optimal performance with a minimal size of validation set. Notably, 5% of the validation data suffices to closely approach the performance of utilizing the full validation sets, as well as to outperform Task Arithmetic by a large margin. Additionally, this trend consistently holds regardless of which acquisition function (EI or UCB) is used when $ratio \geq 30\%$. Consequently, DF-Merge can save the data labeling cost and reduce the computations for running inference on the validation sets while keeping its performance intact.

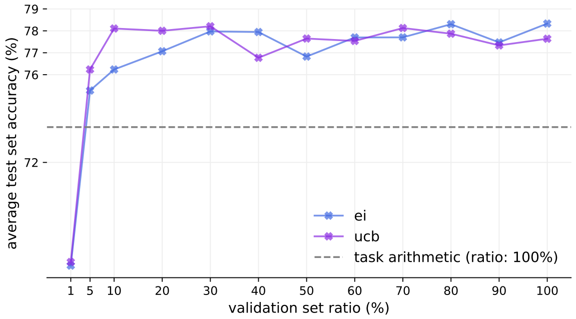

Figure 5: Average test set accuracy (%) of DF-Merge with varying ratios of validation samples used, T5-base, based on a single run.

5.3 Metric Landscape of DF-Merge

We examine how DF-Merge improves multi-task merging by analyzing the metric landscape of General Task Arithmetic and DF-Merge. In particular, we visualize instances of GTA and DF-Merge specified by two merging coefficients in a two-dimensional subspace[4], as shown in Figure 6.

We make the following key observations. **First,**

[4] Following Garipov et al. (2018), we let $u = \tau_1$, $\hat{u} = u/\|u\|$, $v = \tau_2 - \langle \tau_2, \hat{u} \rangle \hat{u}$, and $\hat{v} = v/\|v\|$. Then, $\hat{u}$ and $\hat{v}$ form an orthonormal basis of a 2-D plane, where a coordinate (λ_1, λ_2) specifies a point in the plane as $P(\lambda_1, \lambda_2) = \theta_{\text{pre}} + \lambda_1 \hat{u} + \lambda_2 \hat{v}$.

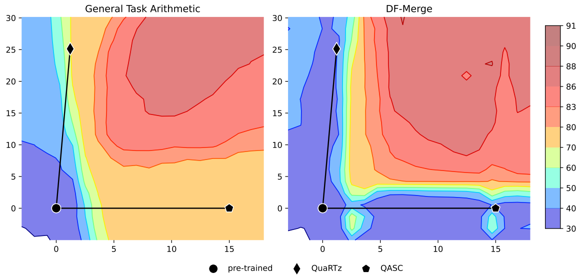

Figure 6: A landscape of average validation accuracy (denoted with colors) of merging two T5-base models, based on a single run. A point in the 2-D plane represents a linear combination of the two task vectors. **left**: GTA; **right**: DF-Merge.

the high accuracy region resides in the upper-right side of the landscape, indicating that the optimal accuracy is unlikely to appear when all coefficients are low. **Second,** GTA and DF-Merge both have a large and flexible search space of the coefficients, compared to TA and Averaging. Overall, DF-Merge has a broader high-accuracy regions compared to GTA, possibly since DF-Merge incorporates *parameter-wise* importance given by Fisher Information. Based on the analysis in Section 2, using Fisher Information can also be interpreted as merging along the low-loss basin. **Third,** DF-Merge underperforms GTA when the coefficients are low, as seen in the lower-right and upper-left side of the landscape. A possible reason is that a model scaled with a low coefficient no longer remains as a local minimum to guarantee the low-loss preserving property of Fisher information.

6 Related Work

Foundations of Model Merging. Recent studies have shown that models sharing the same initialization reside in the same low-loss basin, often connected by a path with non-increasing loss, known as *mode connectivity* (Garipov et al., 2018; Draxler et al., 2018; Mirzadeh et al., 2021). On the contrary, barriers often exists between models optimized from different initialization (Neyshabur et al., 2020). Entezari et al. (2022) shows that SGD solutions from different random initialization can be teleported to the same low-loss basin after accounting for the permutation invariance of neural network. This idea has been introduced to merging models with different initializations (Ainsworth et al., 2023; Stoica et al., 2024). In this paper, we focus on merging fine-tuned models from the same pre-trained initialization.

Building Multitask Model via Merging. An important application of model merging is building a multitask model out of multiple task-specific models fine-tuned from the same initialization (e.g., pre-trained model). While simple averaging is a strong baseline that improves single task merging (Wortsman et al., 2022), it falls significantly short when applied to multitask scenarios. This has led to a series of methods which bridge the gap with multitask fine-tuned models: Fisher Merging (Matena and Raffel, 2022) frames merging as a maximization of joint posterior of models' parameters. RegMean (Jin et al., 2023) minimizes regression errors between the merged model and the fine-tuned models. Unlike previous methods aiming to find closed-form solution, Tam et al. (2024) shows that their iterative method can solve an improved merging objective which is intractable to solve analytically. Meanwhile, Task Arithmetic (TA) (Ilharco et al., 2023) presents a scalable approach for editing fine-tuned parameters to guide the behavior of pre-trained models, a theoretical analysis of which suggests that weight disentanglement arising from pre-training is what makes TA successful (Ortiz-Jimenez et al., 2023). Building on these findings, recent works has explored effective methods for editing fine-tuned parameters with different emphasis, such as resolving parameter interference (Yadav et al., 2024; Daheim et al., 2024), sparsifying task vectors (Yu et al., 2024; Davari and Belilovsky, 2024; Deep et al., 2024), training coefficients (Yang et al., 2024), or applying to adapters (Tang et al., 2024).

Bayesian Optimization in NLP. Bayesian optimization is a family of iterative algorithms for efficient hyperparameter search over a black-box function that is expensive to evaluate. Its applications are found in a range of tasks in NLP, such as optimizing hyperparmeters for text representation (Yogatama et al., 2015), data selection criteria (Ruder and Plank, 2017), and model ensemble (Pour et al., 2024). Most importantly, Liu et al. (2024) utilize Bayesian optimization to find coefficients for average merging that improve checkpoint merging during LLM pre-training. Instead, we leverage Bayesian optimization conditioned on our newly proposed merging objective.

7 Conclusion

In this work, we propose an unified merging framework and introduce Dynamic Fisher-weighted

Merging (**DF-Merge**). This approach assigns scaling coefficients to fine-tuned model parameters and dynamically adjusts them using Bayesian optimization, with the goal of maximizing validation performance. Through this process, DF-Merge tries to efficiently identify low-loss basins using Fisher information. Experimental results demonstrate that DF-Merge consistently outperforms strong baselines across models of different sizes on diverse tasks. The method proves effective in achieving near-optimal performance in just a few iterations, even with minimal validation data, highlighting its potential as a powerful tool for multitask model merging.

Limitations

In this section, we discuss the limitations of our work as follows. **First**, DF-Merge requires the fine-tuned model share the same architecture and pre-trained parameters. Though DF-Merge covers a majority of merging settings given the prevalence of fine-tuning the same pre-trained model, there indeed exist scenarios where one wish to fuse the distinct task expertise of models with different initializations or even across incompatible architectures. We leave this direction for the future research. **Second**, DF-Merge relies on the labeled validation sets, albeit with a relatively small number of samples required to achieve optimal performance. Yet we believe there may be ways to apply DF-Merge when the validation sets are not available. For instance, the fine-tuned models could serve as the pseudo-labeler at the test time, in which case the merging objective becomes maximally replicating each model's predictions on the test inputs. Since we do not leverage the label when estimating the Fisher information, the above approach is feasible. **Third**, our use of Fisher Information is restricted to its diagonal simplification, as the Fisher Information is intractable to compute given its extremely large number of entries ($O(d^2)$ with d being the number of model parameters) for modern PLMs. Diagonal Fisher information implicitly supposes the model parameters are not related to each other in terms of gradient, which is a strong assumption that might lead to suboptimal performance. A promising research direction would be relaxing this assumption, such as representing Fisher Information as a block-diagonal matrix (Tam et al., 2024).

Ethics Statement

Potential Risks If some of the fine-tuned models are trained for malicious purpose, then the merged model DF-Merge might risk producing biased predictions, harmful contents, or unfair decisions, even if the safety of other models are guaranteed. Our method does not address these potential risks, therefore the safety of the merged mode must be checked before deployment.

Use of Scientific Artifacts The **Bayesian Optimization** (Nogueira, 2014–) is under MIT license and **PromptSource** (Bach et al., 2022) is under Apache-2.0 license, both of which permits the use of the tool for research purpose. For the datasets used in our experiments, **PAWS** (Zhang et al., 2019) permits its free use for any purpose, QASC (Khot et al., 2020) is under the CC BY 4.0 license, QuaRTz (Tafjord et al., 2019) and Story Cloze (Sharma et al., 2018) are under Creative Commons License, Winogrande (Sakaguchi et al., 2021) is under Apache, and WikiQA (Yang et al., 2015) is licensed under Microsoft Research Data License Agreement for Microsoft Research WikiQA Corpus. These datasets are publicly available for research purpose. The datasets are intended to serve as benchmarks for testing the ability of AI models on language tasks, hence our experiments are aligned with the intended use.

Model Size and Computational Budget We use T5-base and T5-large (Raffel et al., 2020) which have 223 million and 738 million parameters, respectively. DF-Merge is cost-effective compared to training models, where running a single iteration of DF-Merge approximately requires 70 seconds for T5-base and 170 seconds for T5-large on a single A100 GPU.

References

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. 2023. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Févry, et al. 2022. Promptsource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Nico Daheim, Thomas Möllenhoff, Edoardo Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. 2024. Model merging by uncertainty-based gradient matching. In *The Twelfth International Conference on Learning Representations*.

MohammadReza Davari and Eugene Belilovsky. 2024. Model breadcrumbs: Scaling multi-task model merging with sparse masks. *Preprint*, arXiv:2312.06795.

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

Yang Deng, Yuexiang Xie, Yaliang Li, Min Yang, Nan Du, Wei Fan, Kai Lei, and Ying Shen. 2019. Multi-task learning with multi-view attention for answer selection and knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6318–6325.

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

Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. 2018. Essentially no barriers in neural network energy landscape. In *International conference on machine learning*, pages 1309–1318. PMLR.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. 2022. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations*.

Peter I Frazier. 2018. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. 2018. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Advances in neural information processing systems*, 31.

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

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

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*, volume 34, pages 8082–8090.

Deyuan Liu, Zecheng Wang, Bingning Wang, Weipeng Chen, Chunshan Li, Zhiying Tu, Dianhui Chu, Bo Li, and Dianbo Sui. 2024. Checkpoint merging via bayesian optimization in llm pretraining. *arXiv preprint arXiv:2403.19390*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. 2021. Linear mode connectivity in multitask and continual learning. In *9th International Conference on Learning Representations, ICLR 2021*.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. 2020. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523.

Fernando Nogueira. 2014–. Bayesian Optimization: Open source constrained global optimization tool for Python.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. 2023. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Mohammad Mahdi Abdollah Pour, Ali Pesaranghader, Eldan Cohen, and Scott Sanner. 2024. Gaussian process optimization for adaptable multi-objective text generation using linearly-weighted language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1529–1536.

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*, 21(140):1–67.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. 2018. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757, Melbourne, Australia. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany. Association for Computational Linguistics.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. 2010. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 1015–1022, Madison, WI, USA. Omnipress.

George Stoica, Daniel Bolya, Jakob Brandt Bjorner, Pratik Ramesh, Taylor Hearn, and Judy Hoffman. 2024. Zipit! merging models from different tasks without training. In *The Twelfth International Conference on Learning Representations*.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. QuaRTz: An open-domain dataset of qualitative relationship questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5941–5946, Hong Kong, China. Association for Computational Linguistics.

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

Anke Tang, Li Shen, Yong Luo, Yibing Zhan, Han Hu, Bo Du, Yixin Chen, and Dacheng Tao. 2024. Parameter-efficient multi-task model fusion with partial linearization. In *The Twelfth International Conference on Learning Representations*.

Christopher KI Williams and Carl Edward Rasmussen. 2006. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

T Wolf. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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 *International conference on machine learning*, pages 23965–23998. PMLR.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36.

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

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Dani Yogatama, Lingpeng Kong, and Noah A Smith. 2015. Bayesian optimization of text representations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2100–2105.

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

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhanpeng Zhou, Zijun Chen, Yilan Chen, Bo Zhang, and Junchi Yan. 2024. On the emergence of cross-task linearity in pretraining-finetuning paradigm. In *Forty-first International Conference on Machine Learning*.

A Training & Testing Details of T5 Models

Training is conducted with a batch size of 64, using the AdamW (Loshchilov and Hutter, 2019) optimizer, a fixed learning rate of 1×10^{-4}, and 2,500 steps for each task-specific model, while the multitask model is trained for 25,000 steps with early stopping. The model with the lowest validation loss is selected for testing. For each test input, we forward the input/output pairs of all possible labels to the model and select the one with the lowest perplexity as the final prediction.

B Optimization Trajectories

We complement the optimization trajectories of DF-Merge for T5-base + EI, T5-base + UCB and T5-large + EI in Figure 7, Figure 9 and Figure 8, respectively.

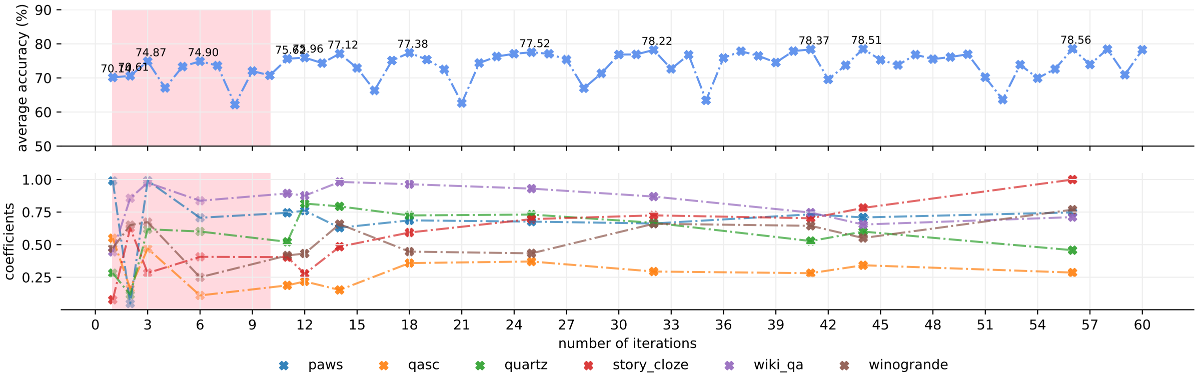

Figure 7: Bayesian optimization trajectory of DF-Merge (T5-base, EI).

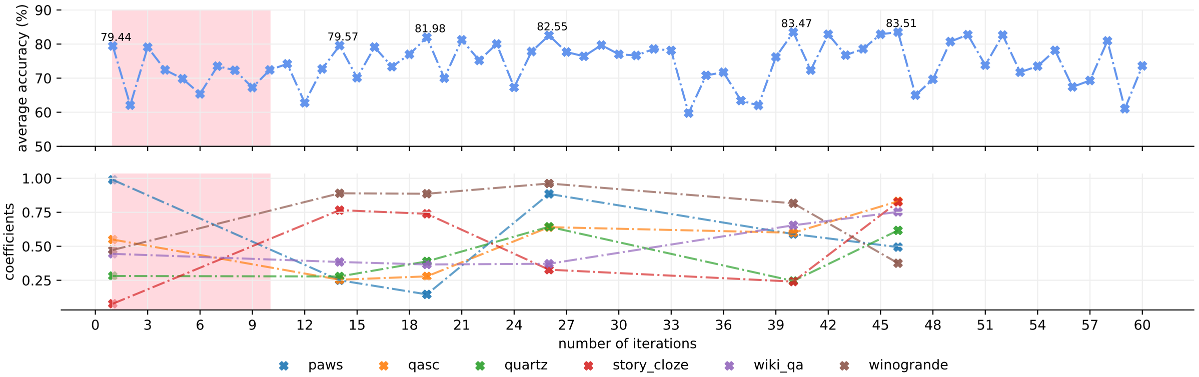

Figure 8: Bayesian optimization trajectory of DF-Merge (T5-large, EI).

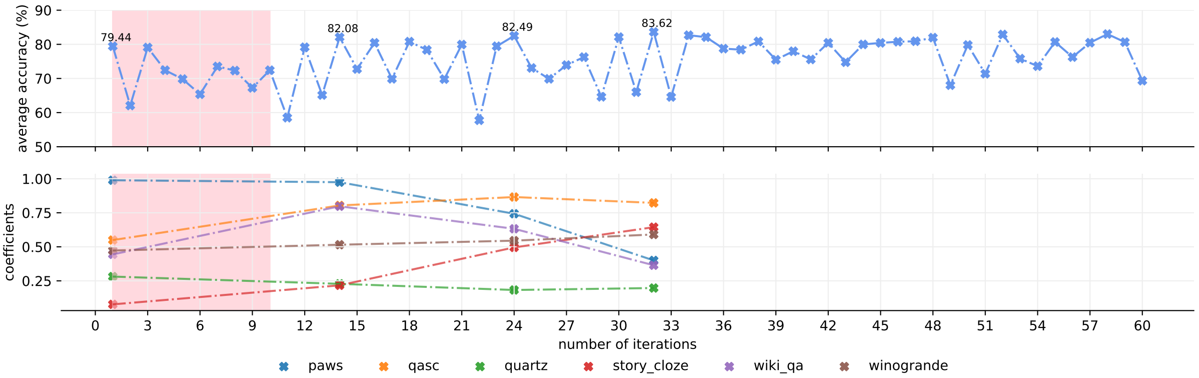

Figure 9: Bayesian optimization trajectory of DF-Merge (T5-large, UCB).