# BiDoRA: Bi-level Optimization-Based Weight-Decomposed Low-Rank Adaptation for Overfitting-Resilient Fine-Tuning of Biological Foundation Models

**Peijia Qin** **Ruiyi Zhang** **Pengtao Xie**
University of California, San Diego
{pqin,ruz048,p1xie}@ucsd.edu

## Abstract

Biological foundation models (e.g., protein language models) are typically fine-tuned on small and noisy datasets, making overfitting a central challenge. We present **BiDoRA**, an overfitting -resilient parameter-efficient fine-tuning (PEFT) method tailored for foundation models. BiDoRA builds on weight-decomposed low-rank adaptation (DoRA) but addresses its over-expressiveness by *decoupling* magnitude and direction optimization within a *bi-level optimization* (BLO) framework: the direction is learned on a training split with magnitudes fixed, while magnitudes are updated on a validation split via hypergradient descent. This design reduces overfitting and yields update patterns that better mimic full fine-tuning under the same parameter budget. On a broad suite of biological and natural language tasks, BiDoRA matches or surpasses strong PEFT baselines. Code is available at https://github.com/t2ance/BiDoRA

## 1 Introduction

Biological foundation models (such as protein language models) are often adapted to tasks where labeled data is scarce and noisy, making overfitting particularly severe. Parameter-efficient fine-tuning (PEFT) methods (Houlsby et al., 2019; Hu et al., 2022b) adapt foundation models—including biological foundation models and LLMs by updating only a small subset of parameters, achieving performance close to full fine-tuning (FT) at much lower cost. A leading PEFT approach is low-rank adaptation (LoRA, Hu et al. (2022b)), which adds and updates low-rank matrices on top of pre-trained weights. Liu et al. (2024a) build on LoRA with DoRA, which explicitly decomposes each weight matrix into magnitude and direction. While this increases expressiveness, it also adds parameters and can worsen overfitting, especially in biological settings with limited data. Moreover, DoRA optimizes both components simultaneously, coupling their updates and constraining the learning pattern.

Our goal is an *overfitting-resilient* PEFT approach that preserves parameter efficiency while improving generalization on small biological datasets. We propose BiDoRA, a **Bi**-level Optimization-Based Weight-**D**ecomposed L**o**w-**R**ank **A**daptation method for PEFT. BiDoRA uses a *bi-level optimization (BLO)* framework to decouple optimization of the two components: the direction is updated on the training split with a tentatively fixed magnitude, while the magnitude is updated on the validation split via hypergradient descent. These steps alternate until convergence. This approach reduces overfitting and enables more flexible updates that better track FT. Fig. 1 provides an overview of BiDoRA. Our design is inspired by DARTS-style NAS that optimizes architecture and weights on disjoint splits; see the discussion in Section I for details. Intuitively, we treat the magnitude vector

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: NeurIPS 2025 2nd Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences.

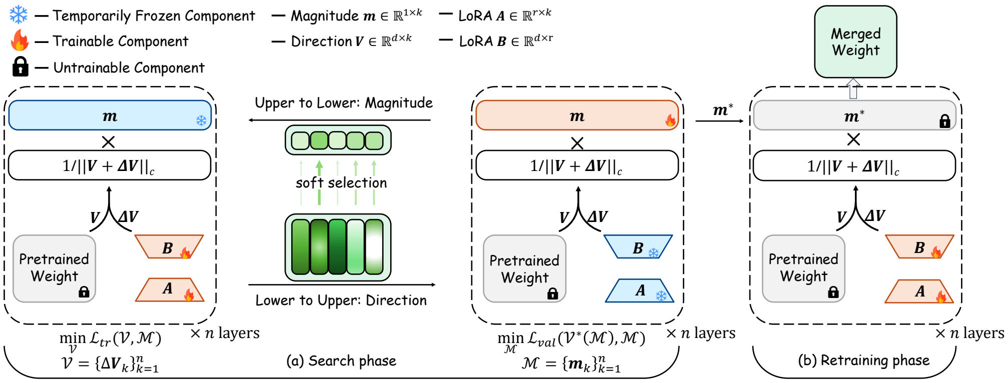

— Temporarily Frozen Component
— Trainable Component
— Untrainable Component
— Magnitude $m \in \mathbb{R}^{1\times k}$
— Direction $V \in \mathbb{R}^{d\times k}$
— LoRA $A \in \mathbb{R}^{r\times k}$
— LoRA $B \in \mathbb{R}^{d\times r}$
Merged Weight
$m$
$1/||V + \Delta V||_c$
$V$
$\Delta V$
Pretrained Weight
$B$
$A$
Upper to Lower: Magnitude
soft selection
Lower to Upper: Direction
$m^*$
$\min_{\mathcal{V}} \mathcal{L}_{tr}(\mathcal{V}, \mathcal{M})$
$\mathcal{V} = \{\Delta V_k\}_{k=1}^n$
$\times n$ layers
(a) Search phase
$\min_{\mathcal{M}} \mathcal{L}_{val}(\mathcal{V}^*(\mathcal{M}), \mathcal{M})$
$\mathcal{M} = \{m_k\}_{k=1}^n$
(b) Retraining phase

Figure 1: **Overview of BiDoRA.** The downstream training data is split into an inner training split $\mathcal{D}_{tr}$ and an outer validation split $\mathcal{D}_{val}$. *Search phase* : the **lower level** updates the direction components $\mathcal{V}$ (low-rank LoRA-style updates $\Delta\mathbf{V} = \mathbf{BA}$) by minimizing the training loss $\mathcal{L}_{tr}$ on $\mathcal{D}_{tr}$ with the orthogonality regularizer, while keeping magnitudes $\mathcal{M}$ fixed; the **upper level** updates $\mathcal{M}$ by descending the hypergradient of the validation loss $\mathcal{L}_{val}$ on $\mathcal{D}_{val}$, computed via a one-step unrolled inner update and a finite-difference approximation. These steps alternate until convergence to yield $\mathcal{M}^*$ . *Retraining phase*: with $\mathcal{M}^*$ fixed, the direction $\mathcal{V}$ is retrained on $\mathcal{D}_{tr} \cup \mathcal{D}_{val}$ to obtain $\mathcal{V}^*$. The final adapted weight is $\mathbf{W}' = \mathbf{m}\frac{\mathbf{W}_0+\mathbf{BA}}{\|\mathbf{W}_0+\mathbf{BA}\|_c}$. Decoupling magnitude (upper) and direction (lower) mitigates overfitting and produces update patterns closer to full fine-tuning.

as an architecture-like selector and the direction matrices as subnetworks; validating magnitudes on held-out data penalizes overfitting and, together with decoupled updates, improves generalization.

Empirically, on biological foundation models, BiDoRA achieves superior performance across a broad suite of protein tasks (Section 4) and NLP benchmarks (Section C). Analyses including weight decomposition, ablation studies, and a train-test gap comparison support the effectiveness of the BLO design, and training efficiency remains competitive.

## 2 Related Work

**Parameter-efficient fine-tuning** (PEFT) methods aim to reduce the high costs associated with full fine-tuning large-scale models by updating only a relatively small subset of pre-trained parameters, rather than the entire model, to adapt to downstream tasks. Existing PEFT methods can be mainly categorized into three types: adapter-based methods (Houlsby et al., 2019; He et al., 2022; Xu et al., 2023; Bi et al., 2024; Yi et al., 2024), prompt tuning methods (Lester et al., 2021; Razdaibiedina et al., 2023), and low-rank adaptation (Hu et al., 2022a; Zhang et al., 2023, 2024b; Kopiczko et al., 2024; Liu et al., 2024b; Gao et al., 2024; Azizi et al., 2024; Shen et al., 2025b,a; Liu et al., 2024a). This work belongs to the third category. **Bi-level optimization** (BLO) has been widely applied in various machine learning tasks, including meta-learning (Finn et al., 2017; Rajeswaran et al., 2019), neural architecture search (NAS) (Liu et al., 2019; Zhang et al., 2021), and hyperparameter optimization (Lorraine et al., 2020; Franceschi et al., 2017). Our method is related to both lines of work as a bilevel-optimization-based low-rank adaptation method. Owing to space constraints, the complete review is provided in Section A.

## 3 Methods

**Preliminaries.** LoRA (Hu et al., 2022b) involves attaching the product of two low-rank matrices to the pre-trained weights and fine-tuning these low-rank matrices on downstream datasets with the pre-trained weights frozen. Formally, given a pre-trained weight matrix $\mathbf{W_0} \in \mathbb{R}^{d\times k}$, LoRA attaches a low-rank update matrix $\Delta\mathbf{W} \in \mathbb{R}^{d\times k}$ to the pre-trained weight. This update matrix can be decomposed as $\Delta\mathbf{W} = \mathbf{BA}$, where $\mathbf{B} \in \mathbb{R}^{d\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times k}$ are two low-rank matrices, with $r \ll \min(d, k)$. Consequently, the weight matrix $\mathbf{W}'$ is represented as $\mathbf{W}' = \mathbf{W_0} + \Delta\mathbf{W} = \mathbf{W_0} + \mathbf{BA}$, with only $\Delta\mathbf{W}$ updated. Liu et al. (2024a) propose weight-decomposed low-rank adaptation (DoRA) to further reparameterize the weight matrices by explicitly decomposing them into learnable magnitude and direction components as $\mathbf{W}' = \mathbf{m}\frac{\mathbf{V}+\Delta\mathbf{V}}{\|\mathbf{V}+\Delta\mathbf{V}\|_c} = \mathbf{m}\frac{\mathbf{W_0}+\mathbf{BA}}{\|\mathbf{W_0}+\mathbf{BA}\|_c}$, where $\Delta\mathbf{V}$ is a product of two learnable low-rank matrices, $\mathbf{B}$ and $\mathbf{A}$, while the magnitude component $\mathbf{m} \in \mathbb{R}^{1\times k}$ is a learnable vector. Here, $\|\cdot\|_c$ represents the vector-wise norm of a matrix computed across each column, using the $L_2$ norm.

**Overview.** BiDoRA optimizes the trainable parameters in DoRA layers by solving a BLO problem. Let $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_n\}$ denote the set of magnitude components for all $n$ DoRA modules, and $\mathcal{V} = \{\Delta\mathbf{V}_1, \Delta\mathbf{V}_2, \ldots, \Delta\mathbf{V}_n\}$ denote the set of corresponding direction components. Specifically, we first learn the direction components $\mathcal{V}^*(\mathcal{M})$ on the training split of the downstream dataset $\mathcal{D}_{tr}$ at the lower level. The magnitude component $\mathcal{M}$ is tentatively fixed at this level; thus, the resulting optimal direction component $\mathcal{V}^*(\mathcal{M})$ is a function of $\mathcal{M}$. At the upper level, we determine the optimal magnitude component $\mathcal{M}^*$ by optimizing the loss on a validation split $\mathcal{D}_{val}$. In practice, $\mathcal{D}_{tr}$ and $\mathcal{D}_{val}$ are typically created by splitting the original training set without using additional data. This BLO problem is solved using an efficient gradient-based algorithm, where parameters at two levels are optimized iteratively until convergence.

**Orthogonal regularization.** To increase the expressiveness of the low-rank direction component $\Delta\mathbf{V}$ and mitigate overfitting, we encourage its columns to be diverse. We impose a Gram regularizer (Xie et al., 2017):

$$\mathcal{R}(\mathcal{V}) = \sum_{k=1}^{n} \left\| (\mathbf{V}_k + \Delta\mathbf{V}_k)^\top (\mathbf{V}_k + \Delta\mathbf{V}_k) - \mathbf{I} \right\|_F^2 \tag{1}$$

where $\mathbf{I}$ is the identity and $\|\cdot\|_F$ the Frobenius norm. With unit-normalized columns, $\mathcal{R}(\mathcal{V})$ promotes (near-)orthogonal, non-redundant directions, improving generalization (see Table 13).

**Lower level.** At the lower level, we train the low-rank direction component $\mathcal{V}$ by minimizing a loss $\mathcal{L}_{tr}$ defined on the training set $\mathcal{D}_{tr}$. The overall training objective at this level is $\mathcal{L}_{tr}(\mathcal{V}, \mathcal{M}) = \mathcal{L}(\mathcal{V}, \mathcal{M}; \mathcal{D}_{tr}) + \gamma\mathcal{R}(\mathcal{V})$. Here, $\mathcal{L}$ represents the fine-tuning loss, given the low-rank direction component $\mathcal{V}$, the magnitude component $\mathcal{M}$, and the training split $\mathcal{D}_{tr}$ of the downstream dataset. $\mathcal{R}(\mathcal{V})$ is the orthogonal regularizer defined in Eq. (1), with $\gamma$ as a trade-off hyperparameter. In this level, we only update $\mathcal{V}$ while keeping $\mathcal{M}$ fixed, resulting in the following optimization problem:

$$\mathcal{V}^*(\mathcal{M}) = \arg\min_{\mathcal{V}} \ \mathcal{L}_{tr}(\mathcal{V}, \mathcal{M}) \tag{2}$$

where $\mathcal{V}^*(\mathcal{M})$ denotes the optimal solution for $\mathcal{V}$ in this problem, which is a function of $\mathcal{M}$.

**Upper level.** At the upper level, we validate the previously fixed magnitudes $\mathcal{M}$ on the validation set $\mathcal{D}_{val}$, using the optimal direction component $\mathcal{V}^*(\mathcal{M})$ that was learned at the lower level. This results in a validation loss $\mathcal{L}_{val}(\mathcal{V}^*(\mathcal{M}), \mathcal{M}) = \mathcal{L}(\mathcal{V}^*(\mathcal{M}), \mathcal{M}; \mathcal{D}_{val})$. We determine the optimal magnitude component $\mathcal{M}$ by minimizing this validation loss:

$$\min_{\mathcal{M}} \mathcal{L}_{val}(\mathcal{V}^*(\mathcal{M}), \mathcal{M}) \tag{3}$$

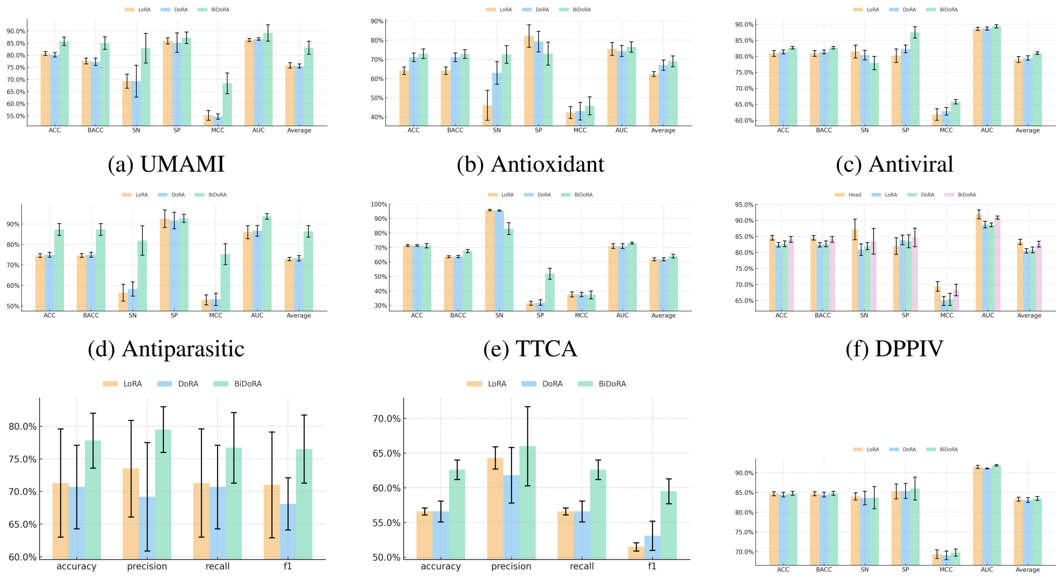
**A bi-level optimization framework.** Integrating the two levels of optimization problems, we have the following BLO framework:

$$\begin{aligned} & \min_{\mathcal{M}} \ \mathcal{L}_{val}(\mathcal{V}^*(\mathcal{M}), \mathcal{M}) \\ s.t. \quad & \mathcal{V}^*(\mathcal{M}) = \arg\min_{\mathcal{V}} \ \mathcal{L}_{tr}(\mathcal{V}, \mathcal{M}) \end{aligned} \tag{4}$$

Note that these two levels of optimization problems are mutually dependent on each other. The solution of the optimization problem at the lower level, $\mathcal{V}^*(\mathcal{M})$, serves as a parameter for the upper-level problem, while the optimization variable $\mathcal{M}$ at the upper level acts as a parameter for the lower-level problem. By solving these two interconnected problems jointly, we can learn the optimal magnitude component $\mathcal{M}^*$ and incremental direction matrices $\mathcal{V}^*$ in an end-to-end manner.

**Optimization algorithm.** We solve the BLO with gradient-based updates (Choe et al., 2023b). Computing the exact hypergradient $\nabla_{\mathcal{M}}\mathcal{L}_{val}(\mathcal{V}^*(\mathcal{M}), \mathcal{M})$ is intractable because it would require fully solving the non-convex inner problem at every step. We therefore adopt a one-step unrolled approximation (Liu et al., 2019):

$$\nabla_{\mathcal{M}}\mathcal{L}_{val}(\mathcal{V}^*(\mathcal{M}), \mathcal{M}) \approx \nabla_{\mathcal{M}}\mathcal{L}_{val}(\mathcal{V} - \xi\nabla_{\mathcal{V}}\mathcal{L}_{tr}(\mathcal{V}, \mathcal{M}), \mathcal{M}) \tag{5}$$



(a) UMAMI (b) Antioxidant (c) Antiviral

(d) Antiparasitic (e) TTCA (f) DPPIV

(g) Anti-CRISPR (h) Vari-Pred (i) Neuropeptides

Figure 2: Summary of results across nine tasks (mean ± std). A higher value is better for all metrics .

**Algorithm 1:** BiDoRA

```
Input: Training dataset $\mathcal{D}_{tr}$ and validation dataset $\mathcal{D}_{val}$
Initialize trainable magnitude components $\mathcal{M} = \{\mathbf{m}_k\}_{k=1}^n$ and low-rank direction components
  $\mathcal{V} = \{\Delta\mathbf{V}_k\}_{k=1}^n = \{\{\mathbf{A}_k\}_{k=1}^n, \{\mathbf{B}_k\}_{k=1}^n\}$
// Search Phase
while not converged do
    Update magnitude $\mathcal{M}$ by descending $\nabla_{\mathcal{M}}\mathcal{L}_{val}(\mathcal{V} - \xi\nabla_{\mathcal{V}}\mathcal{L}_{tr}(\mathcal{V}, \mathcal{M}), \mathcal{M})$
    Update direction $\mathcal{V}$ by descending $\nabla_{\mathcal{V}}\mathcal{L}_{tr}(\mathcal{V}, \mathcal{M})$
Obtain the optimal magnitude $\mathcal{M}^* = \{m_k^*\}_{k=1}^n$
// Retraining Phase
Train $\mathcal{V}$ until convergence using $\mathcal{D}_{tr} \bigcup \mathcal{D}_{val}$ and derive the optimal direction $\mathcal{V}^*$
Output: $\mathcal{V}^*$ and $\mathcal{M}^*$
```

where $\xi$ is the learning rate at the lower level, and the one-step-unrolled model $\bar{\mathcal{V}} = \mathcal{V} - \xi\nabla_{\mathcal{V}}\mathcal{L}_{tr}(\mathcal{V}, \mathcal{M})$ is used as a surrogate for the optimal solution $\mathcal{V}^*(\mathcal{M})$. We then compute the approximated gradient as follows:

$$
\begin{aligned}
&\nabla_{\mathcal{M}}\mathcal{L}_{val}(\mathcal{V} - \xi\nabla_{\mathcal{V}}\mathcal{L}_{tr}(\mathcal{V}, \mathcal{M}), \mathcal{M}) \\
=&\nabla_{\mathcal{M}}\mathcal{L}_{val}(\bar{\mathcal{V}}, \mathcal{M}) - \xi\nabla^2_{\mathcal{M},\mathcal{V}}\mathcal{L}_{tr}(\mathcal{V}, \mathcal{M})\nabla_{\bar{\mathcal{V}}}\mathcal{L}_{val}(\bar{\mathcal{V}}, \mathcal{M}) && (6) \\
\approx&\nabla_{\mathcal{M}}\mathcal{L}_{val}(\bar{\mathcal{V}}, \mathcal{M}) - \xi\frac{\nabla_{\mathcal{M}}\mathcal{L}_{tr}(\mathcal{V}^+, \mathcal{M}) - \nabla_{\mathcal{M}}\mathcal{L}_{tr}(\mathcal{V}^-, \mathcal{M})}{2\epsilon} && (7)
\end{aligned}
$$

Here $\epsilon$ is small and $\mathcal{V}^{\pm} = \mathcal{V} \pm \epsilon\nabla_{\bar{\mathcal{V}}}\mathcal{L}_{val}(\bar{\mathcal{V}}, \mathcal{M})$. The Hessian–vector product in Eq. (6) is approximated via finite differences as in Eq. (7). We alternate gradient steps on $\mathcal{M}$ (validation) and $\mathcal{V}$ (training) until convergence, then retrain $\mathcal{V}$ on $\mathcal{D}_{tr} \cup \mathcal{D}_{val}$ with $\mathcal{M}$ fixed to $\mathcal{M}^*$.

We set magnitudes as the upper-level variables for two reasons: (1) the upper level typically has far fewer parameters than the lower level—here, $\mathcal{O}(k)$ for magnitudes versus $\mathcal{O}(dr + kr)$ for directions—which aligns with common BLO practice; and (2) the magnitude vector behaves like an architecture-selection variable in DARTS (Liu et al., 2019), softly selecting directional subspaces via scaling.

In practice, the convergence of the search phase is determined by the evaluation metric at the upper level. For the subsequent retraining phase, we adopt a stopping criterion similar to DoRA's, observing performance on a separate, held-out test set that is not used during training.

## 4 Experiments

Table 1: Fine-tuning ESM on the thermostability prediction task (Chen et al., 2023b) (left), the BBP task (Dai et al., 2021) (middle), and the MIC task (Ledesma-Fernandez et al., 2023) (right). A higher value is better for all metrics except for MSE. The best results are highlighted in **bold**.

| Methods | #Params | Accuracy | Precision | Recall | F1 | #Params | Accuracy | Precision | Recall | F1 | #Params | MSE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FT | 652.7M | 79.8 | 81.2 | 79.8 | 78.4 | 652.9M | 89.4 | 89.9 | 89.4 | 89.4 | 652.7M | 0.2894 |
| LoRA | 1.5M | 75.9 | 78.2 | 75.9 | 75.5 | 1.9M | 86.8 | 87.7 | 86.8 | 86.7 | 1.7M | 0.3433 |
| DoRA | 1.6M | 76.9 | 78.7 | 76.9 | 76.2 | 2.0M | 89.4 | 91.3 | 89.4 | 89.3 | 1.8M | 0.2918 |
| BiDoRA | 1.6M | **78.8** | **79.1** | **78.8** | **78.2** | 2.0M | **92.1** | **93.1** | **92.1** | **92.0** | 1.8M | **0.2818** |

We evaluated BiDoRA across diverse domains, including biological tasks and natural language processing tasks. BiDoRA **does not use any additional data** compared to other baselines, as we create the validation set for upper-level optimization by splitting the original training set with an 8:2 ratio for all tasks. Our implementation is based on the Hugging Face Transformers library (Wolf et al., 2019) and the Betty library (Choe et al., 2023b).

For comprehensive experimental setup, dataset descriptions, baselines, please see Section B.

Table 1 and Fig. 2 present results from fine-tuning ESM, a transformer-based protein language model (Rives et al., 2021), across a wide range of datasets, including Type I anti-CRISPR, pathogenic missense variants, thermostability, blood-brain barrier peptides, umami peptides, antioxidant peptides, antiviral peptides, antiparasitic peptides, tumor T-cell antigens, DPP-IV inhibitory peptides, neuropeptides, and MIC regression.

The results show that BiDoRA achieves superior or comparable performance to strong baselines across all datasets with the same number of trainable parameters. This verifies the effectiveness of the BLO mechanism: by training the magnitude and direction components on two distinct splits, BiDoRA enhances the flexibility of the learning process and improves learning capacity compared to DoRA.

Empirically, BiDoRA reduces overfitting and better matches FT's learning pattern. On GLUE, the average train-test gap drops from 12.9 (DoRA) to 8.5 (BiDoRA) in Table 5. The weight-decomposition analysis in Section D shows that BiDoRA yields stronger negative correlations between magnitude and direction changes—e.g., Query: $-8.042$ (BiDoRA) vs. $-1.784$ (DoRA), Value: $-10.547$ vs. $-5.485$—closer to FT, while LoRA remains positive (Fig. 4). These improvements are statistically significant (Wilcoxon signed-rank test, $p = 2.4 \times 10^{-4}$; Section I), and the overall training cost remains competitive (Section E).

## 5 Conclusion

We presented BiDoRA, a bi-level optimization framework for parameter-efficient fine-tuning that separates the optimization of magnitudes and directions over disjoint data splits. Our analyses suggest two takeaways. First, decoupling reduces overfitting and aligns the learning dynamics more closely with full fine-tuning. Second, the benefits persist across model scales and tasks, and are robust under standard ablations. Beyond achieving strong accuracy, BiDoRA offers a simple training recipe that is compatible with existing PEFT/DoRA implementations and requires no architectural changes, making adoption practical in settings with limited data.

## References

Seyedarmin Azizi, Souvik Kundu, and Massoud Pedram. Lamda: Large model fine-tuning via spectrally decomposed low-dimensional adaptation. *ArXiv preprint*, abs/2406.12832, 2024. URL https://arxiv.org/abs/2406.12832.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss (eds.), *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for*

*Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, 2005. Association for Computational Linguistics. URL https://aclanthology.org/W05-0909.

Fan Bao, Guoqiang Wu, Chongxuan Li, Jun Zhu, and Bo Zhang. Stability and generalization of bilevel programming in hyperparameter optimization. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 4529–4541, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/2406a0a94c80406914ff2f6c9fdd67d5-Abstract.html.

Qi Bi, Jingjun Yi, Hao Zheng, Haolan Zhan, Yawen Huang, Wei Ji, Yuexiang Li, and Yefeng Zheng. Learning frequency-adapted vision foundation model for domain generalized semantic segmentation. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/aaf50c91c3fc018f6a476032d02114d9-Abstract-Conference.html.

Yannan Bin, Wei Zhang, Wending Tang, Ruyu Dai, Menglu Li, Qizhi Zhu, and Junfeng Xia. Prediction of neuropeptides from sequence information using ensemble classifier and hybrid features. *Journal of proteome research*, 19(9):3732–3740, 2020.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens (eds.), *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2001. URL https://aclanthology.org/S17-2001.

Phasit Charoenkwan, Sakawrat Kanthawong, Chanin Nantasenamat, Md Mehedi Hasan, and Watshara Shoombuatong. idppiv-scm: a sequence-based predictor for identifying and analyzing dipeptidyl peptidase iv (dpp-iv) inhibitory peptides using a scoring card method. *Journal of proteome research*, 19(10):4125–4136, 2020a.

Phasit Charoenkwan, Chanin Nantasenamat, Md Mehedi Hasan, and Watshara Shoombuatong. ittca-hybrid: Improved and robust identification of tumor t cell antigens by utilizing hybrid feature representation. *Analytical biochemistry*, 599:113747, 2020b.

Phasit Charoenkwan, Janchai Yana, Chanin Nantasenamat, Md Mehedi Hasan, and Watshara Shoombuatong. iumami-scm: a novel sequence-based predictor for prediction and analysis of umami peptides using a scoring card method with propensity scores of dipeptides. *Journal of Chemical Information and Modeling*, 60(12):6666–6678, 2020c.

Tianlong Chen, Chengyue Gong, Daniel J Diaz, Xuxi Chen, Jordan T Wells, Qiang Liu, Zhangyang Wang, Andrew D Ellington, Alexandros G Dimakis, and Adam Klivans. Hotprotein: A novel framework for protein thermostability prediction and editing. ICLR 2023 https://openreview.net/forum? id= YDJRFWBMNby, 2023a.

Tianlong Chen, Chengyue Gong, Daniel Jesus Diaz, Xuxi Chen, Jordan Tyler Wells, Qiang Liu, Zhangyang Wang, Andrew D. Ellington, Alex Dimakis, and Adam R. Klivans. Hotprotein: A novel framework for protein thermostability prediction and editing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023b. URL https://openreview.net/pdf?id=YDJRFWBMNby.

Jun Cheng, Guido Novati, Joshua Pan, Clare Bycroft, Akvilė Žemgulytė, Taylor Applebaum, Alexander Pritzel, Lai Hong Wong, Michal Zielinski, Tobias Sargeant, et al. Accurate proteome-wide missense variant effect prediction with alphamissense. *Science*, 381(6664):eadg7492, 2023.

Sang Keun Choe, Sanket Vaibhav Mehta, Hwijeen Ahn, Willie Neiswanger, Pengtao Xie, Emma Strubell, and Eric P. Xing. Making scalable meta learning practical. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December*

*10 - 16, 2023*, 2023a. URL http://papers.nips.cc/paper_files/paper/2023/hash/531998dc1fc858b5857a90b74d96ecab-Abstract-Conference.html.

Sang Keun Choe, Willie Neiswanger, Pengtao Xie, and Eric P. Xing. Betty: An automatic differentiation library for multilevel optimization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023b. URL https://openreview.net/pdf?id=LV_MeMS38Q9.

Nigel Collier, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Jin-Dong Kim. Introduction to the bio-entity recognition task at JNLPBA. In Nigel Collier, Patrick Ruch, and Adeline Nazarenko (eds.), *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pp. 73–78, Geneva, Switzerland, 2004. COLING. URL https://aclanthology.org/W04-1213.

Hua Cui and Jie Bai. A new hyperparameters optimization method for convolutional neural networks. *Pattern Recognition Letters*, 125:828–834, 2019.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pp. 177–190. Springer, 2005.

Ruyu Dai, Wei Zhang, Wending Tang, Evelien Wynendaele, Qizhi Zhu, Yannan Bin, Bart De Spiegeleer, and Junfeng Xia. Bbppred: sequence-based prediction of blood-brain barrier peptides with feature representation learning and logistic regression. *Journal of Chemical Information and Modeling*, 61(1):525–534, 2021.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://aclanthology.org/I05-5002.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 2017. URL http://proceedings.mlr.press/v70/finn17a.html.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1165–1173. PMLR, 2017. URL http://proceedings.mlr.press/v70/franceschi17a.html.

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete fourier transform. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=XUOHKSsurt.

Moein Hasani, Chantel N. Trost, Nolen Timmerman, and Lingling Jin. Acrtransact: Pre-trained protein transformer models for the detection of type i anti-crispr activities. In *Proceedings of the 14th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 1–6, 2023.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=0RDcd5Axok.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019. URL http://proceedings.mlr.press/v97/houlsby19a.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022a. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022b. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Iurii Kemaev, Dan A Calian, Luisa M Zintgraf, Gregory Farquhar, and Hado van Hasselt. Scalable meta-learning via mixed-mode differentiation. *ArXiv preprint*, abs/2505.00793, 2025. URL https://arxiv.org/abs/2505.00793.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. Vera: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=NjNfLdxr3A.

Alba Ledesma-Fernandez, Susana Velasco-Lozano, Javier Santiago-Arcos, Fernando López-Gallego, and Aitziber L Cortajarena. Engineered repeat proteins as scaffolds to assemble multi-enzyme systems for efficient cell-free biosynthesis. *Nature Communications*, 14(1):2587, 2023.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pp. 605–612, Barcelona, Spain, 2004. doi: 10.3115/1218955.1219032. URL https://aclanthology.org/P04-1077.

Weining Lin, Jude Wells, Zeyuan Wang, Christine Orengo, and Andrew C. R. Martin. Enhancing missense variant pathogenicity prediction with protein language models using VariPred. *Scientific Reports*, 14(1):8136, 2024.

Zhaojiang Lin, Andrea Madotto, and Pascale Fung. Exploring versatile generative language model via parameter-efficient transfer learning. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 441–459, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.41. URL https://aclanthology.org/2020.findings-emnlp.41.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=S1eYHoC5FX.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=3d5CIRG1n2.

Zeyu Liu, Souvik Kundu, Anni Li, Junrui Wan, Lianghao Jiang, and Peter Anthony Beerel. Aflora: Adaptive freezing of low rank adaptation in parameter efficient fine-tuning of large models. *ArXiv preprint*, abs/2403.13269, 2024b. URL https://arxiv.org/abs/2403.13269.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In Silvia Chiappa and Roberto Calandra (eds.), *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1540–1552. PMLR, 2020. URL http://proceedings.mlr.press/v108/lorraine20a.html.

Amram Mor. Multifunctional host defense peptides: antiparasitic activities. *The FEBS journal*, 276 (22):6474–6482, 2009.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. The E2E dataset: New challenges for end-to-end generation. In Kristiina Jokinen, Manfred Stede, David DeVault, and Annie Louis (eds.), *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 201–206, Saarbrücken, Germany, 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5525. URL https://aclanthology.org/W17-5525.

Tobias Hegelund Olsen, Betül Yesiltas, Frederikke Isa Marin, Margarita Pertseva, Pedro J García-Moreno, Simon Gregersen, Michael Toft Overgaard, Charlotte Jacobsen, Ole Lund, Egon Bech Hansen, et al. Anoxpepred: using deep learning for the prediction of antioxidative properties of peptides. *Scientific reports*, 10(1):21471, 2020.

Divya Padmanabhan, Satyanath Bhat, Shirish Shevade, and Y Narahari. Topic model based multi-label classification. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 996–1003. IEEE, 2016.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Barak A Pearlmutter and Jeffrey Mark Siskind. Reverse-mode ad in a functional framework: Lambda the ultimate backpropagator. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 30(2):1–36, 2008.

Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 737–746. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/pedregosa16.html.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.39. URL https://aclanthology.org/2021.eacl-main.39.

Sergio A Pinacho-Castellanos, César R García-Jacas, Michael K Gilson, and Carlos A Brizuela. Alignment-free antimicrobial peptide predictors: improving performance by a thorough analysis of the largest available data set. *Journal of Chemical Information and Modeling*, 61(6):3141–3157, 2021.

Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. Meta-learning with implicit gradients. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 113–124, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/072b030ba126b2f4b2374f342be9ed44-Abstract.html.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124.

Hanne B Rasmussen, Sven Branner, Finn C Wiberg, and Nicolai Wagtmann. Crystal structure of human dipeptidyl peptidase iv/cd26 in complex with a substrate analog. *Nature structural biology*, 10(1):19–25, 2003.

Anastasiia Razdaibiedina, Yuning Mao, Madian Khabsa, Mike Lewis, Rui Hou, Jimmy Ba, and Amjad Almahairi. Residual prompt tuning: improving prompt tuning with residual reparameterization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 6740–6757, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.421. URL https://aclanthology.org/2023.findings-acl.421.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. AdapterDrop: On the efficiency of adapters in transformers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7930–7946, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.626. URL https://aclanthology.org/2021.emnlp-main.626.

Yixian Shen, Qi Bi, Jia-Hong Huang, Hongyi Zhu, Andy D Pimentel, and Anuj Pathania. Macp: Minimal yet mighty adaptation via hierarchical cosine projection. *ArXiv preprint*, abs/2505.23870, 2025a. URL https://arxiv.org/abs/2505.23870.

Yixian Shen, Qi Bi, Jia-hong Huang, Hongyi Zhu, Andy D. Pimentel, and Anuj Pathania. SSH: Sparse spectrum adaptation via discrete hartley transformation. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *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)*, pp. 10400–10415, Albuquerque, New Mexico, 2025b. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.522. URL https://aclanthology.org/2025.naacl-long.522/.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002. URL https://aclanthology.org/W02-2024.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 4566–4575. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7299087. URL https://doi.org/10.1109/CVPR.2015.7299087.

Liana Costa Pereira Vilas Boas, Marcelo Lattarulo Campos, Rhayfa Lorrayne Araujo Berlanda, Natan de Carvalho Neves, and Octávio Luiz Franco. Antiviral peptides as promising therapeutic drugs. *Cellular and Molecular Life Sciences*, 76(18):3525–3542, 2019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. In Carles Sierra (ed.), *Proceedings of the Twenty-Sixth International Joint*

*Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 4144–4150. ijcai.org, 2017. doi: 10.24963/ijcai.2017/579. URL https://doi.org/10.24963/ijcai.2017/579.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv preprint*, abs/1910.03771, 2019. URL https://arxiv.org/abs/1910.03771.

Di Xie, Jiang Xiong, and Shiliang Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5075–5084. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.539. URL https://doi.org/10.1109/CVPR.2017.539.

Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 2945–2954. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00288. URL https://doi.org/10.1109/CVPR52729.2023.00288.

Jingjun Yi, Qi Bi, Hao Zheng, Haolan Zhan, Wei Ji, Yawen Huang, Yuexiang Li, and Yefeng Zheng. Learning spectral-decomposited tokens for domain generalized semantic segmentation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 8159–8168, 2024.

Li Zhang, Han Guo, Leah V Schaffer, Young Su Ko, Digvijay Singh, Hamid Rahmani, Danielle Grotjahn, Elizabeth Villa, Michael Gilson, Wei Wang, et al. Proteinaligner: A multi-modal pretraining framework for protein foundation models. *bioRxiv*, pp. 2024–10, 2024a.

Miao Zhang, Steven W. Su, Shirui Pan, Xiaojun Chang, M. Ehsan Abbasnejad, and Reza Haffari. idarts: Differentiable architecture search with stochastic implicit gradients. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12557–12566. PMLR, 2021. URL http://proceedings.mlr.press/v139/zhang21s.html.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=lq62uWRJjiY.

Ruiyi Zhang, Rushi Qiang, Sai Ashish Somayajula, and Pengtao Xie. AutoLoRA: Automatically tuning matrix ranks in low-rank adaptation based on meta learning. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5048–5060, Mexico City, Mexico, 2024b. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.282.

Wei Zhang, Enhua Xia, Ruyu Dai, Wending Tang, Yannan Bin, and Junfeng Xia. Predapp: predicting anti-parasitic peptides with undersampling and ensemble approaches. *Interdisciplinary Sciences: Computational Life Sciences*, 14(1):258–268, 2022.

Yin Zhang, Chandrasekar Venkitasamy, Zhongli Pan, Wenlong Liu, and Liming Zhao. Novel umami ingredients: Umami peptides and their taste. *Journal of food science*, 82(1):16–23, 2017.

Tang-Bin Zou, Tai-Ping He, Hua-Bin Li, Huan-Wen Tang, and En-Qin Xia. The structure-activity relationship of the antioxidant peptides from natural proteins. *Molecules*, 21(1):72, 2016.

# A Related Work

## A.1 Parameter Efficient Fine-Tuning Methods

Parameter-efficient fine-tuning (PEFT) methods aim to reduce the high costs associated with full fine-tuning large-scale models by updating only a relatively small subset of pre-trained parameters, rather than the entire model, to adapt to downstream tasks. Existing PEFT methods can be mainly categorized into three types.

**The first category, known as adapter-based methods,** injects additional trainable modules into the original frozen backbone. For instance, Houlsby et al. (2019) suggests adding linear modules in sequence to existing layers, while He et al. (2022) proposes integrating these modules in parallel with the original layers to enhance performance. Recent advances include SAN (Xu et al., 2023), FADA (Bi et al., 2024), and SET (Yi et al., 2024). SAN presents a side adapter network attached to a frozen CLIP model, which contains two branches for predicting mask proposals and attention biases. FADA introduces a frequency-adapted learning scheme that uses the Haar wavelet transform to decompose frozen features into low- and high-frequency components, which are processed separately to enhance domain generalization. SET proposes a spectral-decomposed token learning framework that leverages the Fast Fourier Transform to separate frozen features into amplitude and phase components, enhancing them with spectral tokens and attention optimization.

**The second category is prompt tuning methods**, which add extra soft tokens (prompts) to the initial input. During the fine-tuning stage, only these trainable soft tokens are updated, as demonstrated in works such as Lester et al. (2021) and Razdaibiedina et al. (2023). Unfortunately, the first two categories lead to increased inference latency compared to fully fine-tuned models.

**The third prominent category focuses on low-rank adaptation**, pioneered by LoRA (Hu et al., 2022a). LoRA injects trainable, low-rank matrices into a model's layers, freezing the original weights. A key advantage is that these low-rank updates can be merged into the original weights before inference, thus incurring no additional latency. Subsequent works have aimed to improve LoRA's efficiency and performance. For instance, AdaLoRA (Zhang et al., 2023) dynamically reallocates the parameter budget based on the importance scores of weight matrices. Zhang et al. (2024b) uses meta-learning to search for the optimal rank of LoRA matrices, further improving its performance on downstream tasks. Pushing parameter efficiency further, VeRA (Kopiczko et al., 2024) employs a single pair of shared low-rank matrices across all layers, while AFLoRA (Liu et al., 2024b) freezes a portion of adaptation parameters based on a learned score. A distinct sub-direction has emerged that performs adaptation in the frequency domain, including FourierFT (Gao et al., 2024), LaMDA (Azizi et al., 2024), SSH (Shen et al., 2025b), and MaCP (Shen et al., 2025a). These methods learn updates in transformed spectral spaces, such as the Fourier, discrete Hartley, or discrete cosine domains, rather than directly in the weight space. Other research has focused on bridging the performance gap between LoRA and full fine-tuning. Liu et al. (2024a) found that LoRA's update patterns differ significantly from full fine-tuning, potentially constraining its learning capacity. To mitigate this, they proposed DoRA (Liu et al., 2024a), which decomposes pre-trained weights into magnitude and direction components and uses LoRA for efficient directional updates, better mimicking full fine-tuning.

## A.2 Bi-level Optimization

Bi-level optimization (BLO) has been widely applied in various machine learning tasks, including meta-learning (Finn et al., 2017; Rajeswaran et al., 2019), neural architecture search (NAS) (Liu et al., 2019; Zhang et al., 2021), and hyperparameter optimization (Lorraine et al., 2020; Franceschi et al., 2017). Despite its wide usage, solving BLO problems can be challenging due to the inherent nature of nested optimization problems. Several algorithms have been proposed to address this challenge, including zeroth-order methods such as Bayesian optimization (Cui & Bai, 2019) and first-order algorithms based on hypergradients (Pearlmutter & Siskind, 2008; Lorraine et al., 2020). Among these approaches, gradient-based BLO has received significant attention because it can scale to high-dimensional problems with a large number of trainable parameters.

Inspired by NAS, where a bi-level approach is used to learn an architecture and its subnetwork weights on separate data splits to prevent overfitting, we adapt the BLO framework to parameter-efficient fine-tuning (PEFT), specifically for the weight-decomposed adaptation introduced by DoRA. Unlike in NAS, where BLO searches for a network architecture, BiDoRA repurposes it to decouple the optimization of a weight matrix's two components: magnitude and direction. This approach marks a significant departure from previous PEFT methods like LoRA and DoRA, which optimize all trainable parameters simultaneously on a single dataset. In this work, we extend the application of gradient-based BLO to develop a robust and effective PEFT method for pre-trained models. By assigning the magnitude and direction components to different optimization levels with distinct data splits, BiDoRA creates a decoupled, flexible updating pattern that better mitigates overfitting and more closely resembles the learning behavior of full fine-tuning.

# B Datasets, Models, and Baselines

In this section, we present the datasets, models, and baselines used in experiments, and summarize the statistical data in Table 2.

## B.1 Datasets and Models

### B.1.1 Natural Language Understanding

The GLUE Benchmark (Wang et al., 2019) comprises a diverse array of tasks that are widely employed for evaluation in natural language understanding. It encompasses two single-sentence classification tasks, three tasks assessing similarity and paraphrasing, and four tasks focusing on natural language inference. Specifically, it includes MNLI (MultiNLI, Williams et al. (2018)), SST-2 (Stanford Sentiment Treebank, Socher et al. (2013)), MRPC (Microsoft Research Paraphrase Corpus, Dolan & Brockett (2005)), CoLA (Corpus of Linguistic Acceptability, Warstadt et al. (2019)), QNLI (Question NLI, Rajpurkar et al. (2018)), QQP (Quora Question Pairs, Wang et al. (2017)), RTE (Recognizing Textual Entailment, Dagan et al. (2005)), and STS-B (Semantic Textual Similarity Benchmark, Cer et al. (2017)). We summarize the statistical data for all datasets within the GLUE Benchmark in Table 2. Following existing practices, the development set is used in GLUE as the test data since the actual test set is not publicly available. We report the overall (matched and mismatched) accuracy for MNLI, Matthew's correlation for CoLA, Pearson correlation for STS-B, and accuracy for the other tasks.

The Reuters-21578 (Padmanabhan et al., 2016) dataset is one of the most widely used data collections for text categorization research. It was collected from the Reuters financial newswire service in 1987 and is used for text classification and natural language processing tasks. Three splits are available: ModApte, ModHayes, and ModLewis. These documents cover various topics, such as politics, economics, and sports. F1 score is used as the evaluation metric across all three splits.

### B.1.2 Natural Language Generation

In our experiments on natural language generation, we use the E2E (Novikova et al., 2017) dataset, which was initially introduced as a dataset for training end-to-end, data-driven natural language generation systems. Multiple references can be associated with each source table used as input. Each sample input $(x, y)$ consists of a series of slot-value pairs accompanied by an associated natural language reference text. The E2E dataset comprises approximately $42$k training examples, $4,600$ validation examples, and $4,600$ test examples from the restaurant domain.

We use the following five evaluation metrics: BLEU (Papineni et al., 2002), NIST (Lin & Och, 2004), METEOR (Banerjee & Lavie, 2005), ROUGE-L (Lin, 2004), and CIDEr (Vedantam et al., 2015).

### B.1.3 Token Classification

For token classification, we fine-tune the RoBERTa-base and RoBERTa-large models on the BioNLP dataset (Collier et al., 2004) and the CoNLL2003 dataset (Tjong Kim Sang, 2002). BioNLP (Collier et al., 2004) is a Named Entity Recognition dataset that contains biological entities such as DNA, RNA, and protein. It is essentially a token classification task where we want to classify each entity in the sequence. CoNLL-2003 (Tjong Kim Sang, 2002) focuses on language-independent named entity recognition. It concentrates on four types of named entities: persons, locations, organizations, and miscellaneous entities that do not belong to the previous three groups. Accuracy, precision, recall, and F1 score are used as evaluation metrics.

Table 2: Summary of datasets used in the experiments

| **Task Group** | **Dataset** | **Metrics** | **Train** | **Dev / Val** | **Test** |
|---|---|---|---|---|---|
| Natural Language Understanding | MNLI | Accuracy | 393k | 20k | 20k |
| | SST-2 | Accuracy | 67k | 872 | 1.8k |
| | MRPC | Accuracy | 3.7k | 408 | 1.7k |
| | CoLA | Matthews Corr | 8.5k | 1k | 1k |
| | QNLI | Accuracy | 108k | 5.7k | 5.7k |
| | QQP | Accuracy | 364k | 40k | 391k |
| | RTE | Accuracy | 2.5k | 276 | 3k |
| | STS-B | Pearson Corr | 7.0k | 1.5k | 1.4k |
| Text Classification | ModApte | F1 | 8.8k | - | 3k |
| | ModHayes | F1 | 18k | - | 0.7k |
| | ModLewis | F1 | 12k | - | 5.5k |
| Natural Language Generation | E2E | BLEU, NIST, MET, ROUGE-L, CIDEr | 42k | 4.6k | - |
| Token Classification | BioNLP | Accuracy, Precision, Recall, F1 | 17k | 1.9k | 3.9k |
| | CoNLL2003 | Accuracy, Precision, Recall, F1 | 14k | 3.3k | 3.5k |
| Biological Experiments | Type I anti-CRISPR (AcrTransAct) | Accuracy, Precision, Recall, F1, AUC | 182 | - | 45 |
| | Pathogenic missense variants (VariPred) | Accuracy, Precision, Recall, F1, AUC | 100 | - | 100 |
| | Thermostability | Accuracy, Precision, Recall, F1, AUC | 936 | - | 104 |
| | Blood-brain barrier peptides (BBP) | ACC, BACC, SN, SP, MCC, AUC | 200 | - | 38 |
| | Umami peptides (UMAMI) | ACC, BACC, SN, SP, MCC, AUC | 353 | - | 89 |
| | Antioxidant peptides (Antioxidant) | ACC, BACC, SN, SP, MCC, AUC | 823 | - | 89 |
| | Antiviral peptides (Antiviral) | ACC, BACC, SN, SP, MCC, AUC | 4,642 | - | 1,246 |
| | Antiparasitic peptides (Antiparasitic) | ACC, BACC, SN, SP, MCC, AUC | 510 | - | 92 |
| | Tumor T-cell antigens (TTCA) | ACC, BACC, SN, SP, MCC, AUC | 788 | - | 197 |
| | DPP-IV inhibitory peptides (DPPIV) | ACC, BACC, SN, SP, MCC, AUC | 1,641 | - | 1,205 |
| | Neuropeptides | ACC, BACC, SN, SP, MCC, AUC | 4,506 | - | 485 |
| | MIC regression (MIC) | MSE | 3,695 | - | 924 |

#### B.1.4 Biological Experiments

We use the ESM (Evolutionary Scale Modeling) model (Rives et al., 2021), a transformer-based protein language model that captures evolutionary patterns in protein sequences through transformer architecture. We fine-tune ESM using the Protein Aligner checkpoint (Zhang et al., 2024a) on diverse datasets, as detailed in the following paragraphs. Protein datasets are typically much smaller than those in NLP, making large pre-trained models prone to overfitting even with parameter-efficient fine-tuning (PEFT) methods. With millions of trainable parameters but only thousands (or hundreds) of available samples, these models are significantly overparameterized, underscoring the value of our overfitting-resilient approach.

**Detection of type I anti-CRISPR activity (Hasani et al., 2023).** This binary classification task determines whether an anti-CRISPR (Acr) protein inhibits a given type I CRISPR-Cas system, using the Acr-CRISPR-Cas inhibition dataset. The benchmark consists of 227 Acr-Cas pairs (132 positive cases of experimentally verified inhibition and 95 negative cases), with performance evaluated on held-out data through binary classification.

**Detection of missense variants pathogenicity (Lin et al., 2024; Cheng et al., 2023).** Given a wild-type protein sequence and its single-amino-acid mutant, this task predicts whether the missense variant is pathogenic or benign. We use a split of 200 labeled examples from VariPred (100 for training, 100 for testing), where labels reflect clinical and curated pathogenicity annotations.

**Prediction of protein thermostability (Chen et al., 2023a).** This task classifies proteins into thermostability categories based on 3D structures. The HP-S2C5 dataset contains 1,040 proteins divided into five temperature ranges: Cryophilic ($-20$-5°C), Psychrophilic (5-25°C), Mesophilic (25-45°C), Thermophilic (45-75°C), and Hyperthermophilic ($>$ 75°C). Following the HotProtein protocol, we use 936 proteins for training and 104 for testing.

**Identification of blood-brain barrier peptides (BBP) (Dai et al., 2021).** This task classifies whether peptides can penetrate the blood-brain barrier. We used the BBPpred dataset, with 100 BBPs and 100 non-BBPs for training, and 19 BBPs and 19 non-BBPs for testing.

**Identification of umami peptides (Charoenkwan et al., 2020c; Zhang et al., 2017).** This task determines whether peptides elicit an umami taste. We used the iUmami-SCM dataset, with 112 umami peptides and 241 non-umami peptides for training, and 28 umami peptides and 61 non-umami peptides for testing.

**Identification of antioxidant peptides (Zou et al., 2016; Olsen et al., 2020).** This task classifies peptides based on their antioxidant properties. We used the AnOxPePred dataset, containing 582 antioxidative peptides and 241 non-antioxidative peptides for training, and 28 antioxidative peptides and 61 non-antioxidative peptides for testing.

**Identification of antiviral peptides (Pinacho-Castellanos et al., 2021; Vilas Boas et al., 2019).** This task predicts whether peptides exhibit antiviral activity against viral infections. We utilized the ABPDiscover dataset, with 2,321 antiviral peptides and 2,321 non-antiviral peptides for training, and 623 antiviral peptides and 623 non-antiviral peptides for testing.

**Identification of antiparasitic peptides (Zhang et al., 2022; Mor, 2009).** This task identifies peptides with antiparasitic activity. Using the PredAPP dataset, we trained on 255 antiparasitic peptides and 255 non-antiparasitic peptides, and tested on 46 antiparasitic peptides and 46 non-antiparasitic peptides.

**Identification of tumor T-cell antigens (TTCA) peptides (Charoenkwan et al., 2020a,b).** This task classifies peptides capable of inducing a T-cell immune response. We used the iTTCA-Hybrid dataset, with 470 antigenic peptides and 318 non-antigenic peptides for training, and 122 antigenic peptides and 75 non-antigenic peptides for testing.

**Identification of dipeptidyl peptidase IV (DPPIV) inhibitory peptides (Rasmussen et al., 2003).** This task identifies peptides that inhibit dipeptidyl peptidase IV (DPP-IV) activity. We used the iDPPIV-SCM dataset, containing 532 inhibitory peptides and 532 non-inhibitory peptides for training, and 133 inhibitory peptides and 133 non-inhibitory peptides for testing.

**Identification of neuropeptides (Bin et al., 2020).** This task classifies peptides as neuropeptides or non-neuropeptides. We used the PredNeuroP dataset, with 1,940 neuropeptides and 1,940 non-neuropeptides for training, and 485 neuropeptides and 485 non-neuropeptides for testing.

**Prediction of the minimum inhibitory concentration of antimicrobial peptides (Ledesma-Fernandez et al., 2023).** This regression task predicts minimum inhibitory concentration (MIC) values from peptide sequences, such as against *E. coli*. We used a curated DeepAMP dataset with 3,695 training examples and 924 testing examples, where labels represent continuous MIC measurements in standard concentration units.

### B.2 Baselines

Here, we provide a brief introduction to compare baselines in all our experiments.

- **Full Fine-Tuning (FT):** The entire model is fine-tuned, with updates to all parameters.
- **Adapter Tuning (Houlsby et al., 2019; Lin et al., 2020; Rücklé et al., 2021; Pfeiffer et al., 2021):** Methods that introduce adapter layers between the self-attention and MLP modules for parameter-efficient tuning.
- **LoRA (Hu et al., 2022a):** A method that estimates weight updates via low-rank matrices.
- **AdaLoRA (Zhang et al., 2023):** An extension of LoRA that dynamically reallocates the parameter budget based on importance scores.
- **DoRA (Liu et al., 2024a):** Decomposes pretrained weights into magnitude and direction, using LoRA for efficient directional updates.
- **VeRA (Kopiczko et al., 2024):** Employs a single pair of low-rank matrices across all layers to reduce trainable parameters.
- **FourierFT (Gao et al., 2024):** Fine-tunes models by learning a subset of spectral coefficients in the Fourier domain.
- **AFLoRA (Liu et al., 2024b):** Freezes low-rank adaptation parameters using a learned score, reducing trainable parameters while maintaining performance.
- **LaMDA (Azizi et al., 2024):** Fine-tunes large models via spectrally decomposed low-dimensional adaptation.
- **SSH (Shen et al., 2025b):** Fine-tunes large models after transforming weight matrices with the discrete Hartley transformation (DHT).
- **MaCP (Shen et al., 2025a):** Fine-tunes large models by projecting the low-rank adaptation weight change into the discrete cosine space.

## C Experiments on Natural Language Processing

### C.1 Experiments on Natural Language Understanding Tasks

In this section, we evaluate the performance of BiDoRA on NLU tasks.

**Main results.** Table 3 presents the results of fine-tuning the RoBERTa-base, RoBERTa-large, and DeBERTa XXL models on the GLUE benchmark with baseline PEFT methods and BiDoRA. The results show that BiDoRA achieves superior or comparable performance compared to baseline methods across all datasets with the same number of trainable parameters. The superior performance of BiDoRA verifies the effectiveness of its BLO mechanism. By training the magnitude and direction components on two distinct splits, BiDoRA enhances the flexibility of the learning process and improves learning capacity compared to DoRA, resulting in a performance boost.

We also evaluate on GLUE with the RoBERTa-base model against a wider set of baselines, following Shen et al. (2025b,a) and citing their reported baseline results for reference. The results in Table 4 indicate that BiDoRA consistently outperforms all baselines, including DoRA, across these diverse NLU tasks, demonstrating its robust generalization capability.

Table 3: RoBERTa$_{\text{base/large}}$ (R$_{\text{b/l}}$) and DeBERTa$_{\text{XXL}}$ (D$_{\text{XXL}}$) with different fine-tuning methods on the GLUE benchmark (Wang et al., 2019). A higher value is better for all datasets. The best results are shown in **bold**.

| Method | #Params | MNLI | SST-2 | MRPC | CoLA | QNLI | QQP | RTE | STS-B | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| R$_{\text{b}}$(FT) | 125.0M | 90.3 | 94.8 | 89.3 | 61.6 | 86.7 | 92.8 | 76.9 | 91.2 | 85.5 |
| R$_{\text{b}}$(Adapter) | 0.9 M | 86.5 | 94.0 | 88.4 | 58.8 | 92.5 | 89.1 | 71.2 | 89.9 | 83.8 |
| R$_{\text{b}}$(LoRA) | 0.15 M | 86.8 | 94.3 | 88.0 | 60.3 | **93.0** | 89.6 | 72.9 | 90.1 | 84.4 |
| R$_{\text{b}}$(DoRA) | 0.17 M | 86.8 | 94.2 | 89.2 | 60.5 | 92.9 | 89.6 | 73.2 | **90.2** | 84.6 |
| R$_{\text{b}}$(BiDoRA) | 0.17 M | **87.1** | **94.4** | **89.4** | **61.3** | 92.7 | **90.6** | **76.0** | 90.1 | **85.2** |
| R$_{\text{l}}$(FT) | 355.0M | 90.2 | 96.4 | 90.9 | 68.0 | 94.7 | 92.2 | 86.6 | 92.4 | 88.9 |
| R$_{\text{l}}$(Adapter) | 0.8M | 90.3 | 96.3 | 87.7 | 66.3 | **94.7** | 91.5 | 72.9 | 91.5 | 86.4 |
| R$_{\text{l}}$(LoRA) | 0.39 M | **90.6** | 96.3 | 90.0 | 66.9 | 94.5 | 91.2 | 86.3 | 91.7 | 88.4 |
| R$_{\text{l}}$(DoRA) | 0.39 M | **90.6** | **96.4** | 89.8 | 65.8 | **94.7** | 91.2 | 86.6 | **92.0** | 88.4 |
| R$_{\text{l}}$(BiDoRA) | 0.39 M | **90.6** | 96.1 | **90.1** | **67.0** | 94.6 | **91.7** | **86.9** | **92.0** | **88.6** |
| D$_{\text{XXL}}$(DoRA) | 4.9M | 91.2 | **96.3** | 92.3 | 71.1 | **95.3** | 91.6 | 91.8 | **90.8** | 90.0 |
| D$_{\text{XXL}}$(BiDoRA) | 4.9M | **91.7** | **96.3** | **92.6** | **72.3** | 95.2 | **92.0** | **92.3** | **90.8** | **90.4** |

Table 4: Performance of various fine-tuning methods on the GLUE benchmark for the RoBERTa-base model. The best ones are highlighted by **bold** and the second ones are highlighted by *italic*.

| Model | SST-2 | MRPC | CoLA | QNLI | RTE | STS-B | Avg. |
|---|---|---|---|---|---|---|---|
| FT | 94.8 | 90.2 | 63.6 | 92.8 | 78.7 | *91.2* | 85.22 |
| BitFit | 93.7 | **92.7** | 62.0 | 91.8 | 81.5 | 90.8 | 85.42 |
| Adpt$^{\text{D}}$ | 94.7 | 88.4 | 62.6 | 93.0 | 75.9 | 90.3 | 84.15 |
| LoRA | *95.1* | 89.7 | 63.4 | *93.3* | 78.4 | **91.5** | 85.23 |
| AdaLoRA | 94.5 | 88.7 | 62.0 | 93.1 | 81.0 | 90.5 | 84.97 |
| AFLoRA | 94.1 | 89.3 | 63.5 | 91.3 | 77.2 | 90.6 | 84.33 |
| LaMDA | 94.6 | 89.7 | 64.9 | 91.7 | 78.2 | 90.4 | 84.92 |
| VeRA | 94.6 | 89.5 | 65.6 | 91.8 | 78.7 | 90.7 | 85.15 |
| FourierFT | 94.2 | 90.0 | 63.8 | 92.2 | 79.1 | 90.8 | 85.02 |
| SSH | 94.1 | *91.2* | 63.6 | 92.4 | 80.5 | 90.9 | 85.46 |
| MaCP | 94.2 | 89.7 | 64.6 | 92.4 | 80.7 | 90.9 | 85.42 |
| DoRA ($r = 8$) | 94.9 | 89.9 | 63.7 | *93.3* | 78.9 | **91.5** | 85.37 |
| BiDoRA ($r = 8$) | **95.7** | 90.2 | *65.8* | **93.4** | 79.4 | 90.5 | 85.83 |
| DoRA ($r = 16$) | 94.8 | 90.4 | 65.6 | 93.1 | *81.9* | 90.7 | *86.08* |
| BiDoRA ($r = 16$) | 95.0 | 90.8 | **66.7** | *93.3* | **82.6** | 90.9 | **86.55** |

Table 6 presents the results of fine-tuning RoBERTa models on the Reuters21578 datasets, a text classification task, where BiDoRA outperforms all baseline methods by an even larger margin. Notably, BiDoRA achieves performance comparable to or even better than full fine-tuning, providing further evidence of its superiority.

**Robustness of BiDoRA towards different rank settings.** We explore the impact of different rank configurations on BiDoRA and DoRA, evaluating them with ranks of 8 and 16 in addition to the rank of 4 used in Table 3. The average accuracies reported in Table 4 demonstrate that BiDoRA consistently surpasses DoRA across all rank configurations, highlighting its resilience and superior performance regardless of the rank setting.

Table 5: Quantitative performance gap between training and test sets for DoRA and BiDoRA using the RoBERTa-base model. The gap is calculated as the training metric minus the test metric, where a smaller value indicates less overfitting.

| Method | SST-2 | MRPC | CoLA | QNLI | RTE | STS-B | Avg. |
|---|---|---|---|---|---|---|---|
| DoRA | 2.0 | 9.5 | 32.5 | 6.6 | 18.0 | 8.8 | 12.9 |
| BiDoRA | **1.7** | **7.0** | **23.3** | **0.2** | **14.0** | **4.7** | **8.5** |

Table 6: RoBERTa$_{base/large}$ ($R_{b/l}$) with different fine-tuning methods on the Reuters21578 (Padmanabhan et al., 2016), BioNLP (Collier et al., 2004), and CoNLL2003 (Tjong Kim Sang, 2002) benchmarks. A higher value is better for all metrics. The best results are shown in **bold**.

| | | Reuters21578 | | | BioNLP | | | | CoNLL2003 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | #Params | ModApte | ModHayes | ModLewis | Accuracy | Precision | Recall | F1 | Accuracy | Precision | Recall | F1 |
| $R_b$(FT) | 125.0M | 85.4 | 77.6 | 77.1 | 93.9 | 69.0 | 78.9 | 73.6 | 99.3 | 95.7 | 96.3 | 96.0 |
| $R_b$(Adapter) | 0.9M | **85.3** | 77.5 | 76.8 | 93.9 | 69.1 | 78.8 | 73.7 | **99.3** | 95.7 | 96.4 | 96.0 |
| $R_b$(LoRA) | 0.15M | 84.7 | 74.3 | 74.7 | 93.9 | 69.0 | 78.8 | 73.6 | **99.3** | 95.4 | 96.3 | 95.8 |
| $R_b$(DoRA) | 0.17M | 84.8 | 78.2 | 76.6 | **94.0** | 69.2 | **79.1** | 73.8 | **99.3** | 95.3 | 96.2 | 95.8 |
| $R_b$(BiDoRA) | 0.17M | **85.3** | **79.9** | **77.6** | 93.9 | **71.2** | 78.6 | **74.7** | **99.3** | **95.9** | **96.5** | **96.2** |
| $R_l$(FT) | 355.0M | 84.8 | 77.5 | 76.6 | 94.0 | 69.4 | 79.6 | 74.1 | 99.4 | 96.2 | 97.0 | 96.6 |
| $R_l$(Adapter) | 0.44M | 84.8 | 77.9 | 76.7 | **94.0** | 69.4 | 79.7 | 74.2 | **99.4** | 96.1 | 97.0 | 96.6 |
| $R_l$(LoRA) | 0.39M | 84.7 | 77.7 | 76.7 | 93.9 | 69.2 | 79.3 | 73.9 | **99.4** | 96.2 | 97.0 | 96.6 |
| $R_l$(DoRA) | 0.39M | 84.8 | 77.4 | 76.7 | **94.0** | 69.4 | **79.7** | 74.2 | **99.4** | 96.2 | **97.1** | 96.6 |
| $R_l$(BiDoRA) | 0.39M | **84.9** | **78.9** | **77.3** | **94.0** | **71.3** | 79.3 | **75.1** | **99.4** | **96.4** | **97.1** | **96.7** |

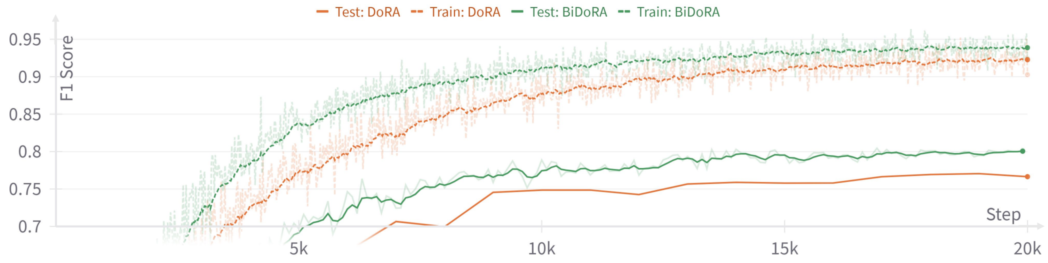


Figure 3: Training and test accuracy versus global training steps on the ModHayes split of the Reuters21578 dataset (Padmanabhan et al., 2016) when fine-tuning a RoBERTa-base model using DoRA and BiDoRA. The training and test curves for DoRA show a larger gap compared to BiDoRA, highlighting the effectiveness of our method in reducing overfitting.

**Performance gap between training and testing set.** As visualized in Fig. 3, BiDoRA achieves a smaller gap between the training and test curves. Quantitatively, Table 5 presents this performance gap on the RoBERTa-base model. The training set metric is calculated as a moving average of the per-batch metric with a decay ratio of $0.99$. Since BiDoRA has two training loops, its training metric is a weighted average ($0.8 \times$ inner-loop-metric $+ \, 0.2 \times$ outer-loop-metric), based on the data split size, inner : outer $= 8 : 2$, in our case. The results show that the performance gap for BiDoRA is consistently lower than that of DoRA across all datasets. This suggests that DoRA is more prone to overfitting, an issue that BiDoRA effectively addresses.

### C.2 Experiments on Natural Language Generation Tasks

In this section, we evaluate BiDoRA's performance on the NLG task. Table 7 presents the results of fine-tuning a GPT-2 model on the E2E dataset with baseline PEFT methods and BiDoRA. The results show that BiDoRA achieves the best performance across all five evaluation metrics, demonstrating the superiority of BiDoRA in fine-tuning pre-trained models for NLG tasks.

Table 7: Performance of BiDoRA and baseline methods for fine-tuning GPT2-medium on the E2E dataset (Novikova et al., 2017). A higher value is better for all metrics. The best results are shown in **bold**.

| Method | #Params | BLEU | NIST | MET | ROUGE-L | CIDEr |
|---|---|---|---|---|---|---|
| FT | 354.9M | 68.0 | 8.61 | 46.1 | 69.0 | 2.38 |
| Adapter | 11.1M | 67.0 | 8.50 | 45.2 | 66.9 | 2.31 |
| LoRA | 0.39M | 67.1 | 8.54 | 45.7 | 68.0 | 2.33 |
| DoRA | 0.39M | 67.0 | 8.48 | 45.4 | 70.1 | 2.33 |
| BiDoRA | 0.39M | **69.0** | **8.72** | **46.2** | **70.9** | **2.44** |

### C.3 Experiments on Token Classification

The effectiveness of BiDoRA can also be observed in Table 6, which reports the results of token classification tasks. Unlike the NLU tasks discussed in the previous section, which involve classifying entire sentences and focusing on capturing global semantics, token classification requires classifying each token within a sentence, highlighting the importance of capturing local context. On the BioNLP dataset, BiDoRA consistently outperforms baseline methods by a large margin in terms of F1 score. On the CoNLL2003 dataset, BiDoRA either outperforms or matches all baseline methods across all metrics. Consistent with our previous findings, BiDoRA effectively fine-tunes pre-trained models for token classification tasks.

### C.4 Experimental Settings

In this section, we provide detailed experimental settings. We maintain consistent configurations across experiments, including LoRA rank, LoRA $\alpha$, batch size, maximum sequence length, and optimizer, to ensure a fair comparison. For results other than Table 4, we do not include the bias term in PEFT linear layers. The hyperparameter tuning for our method is straightforward and convenient.

**RoBERTa** We summarize the experimental settings for the GLUE benchmark (Table 3) and for the Reuters21578 dataset and token classification (Table 6) tasks in Table 8.

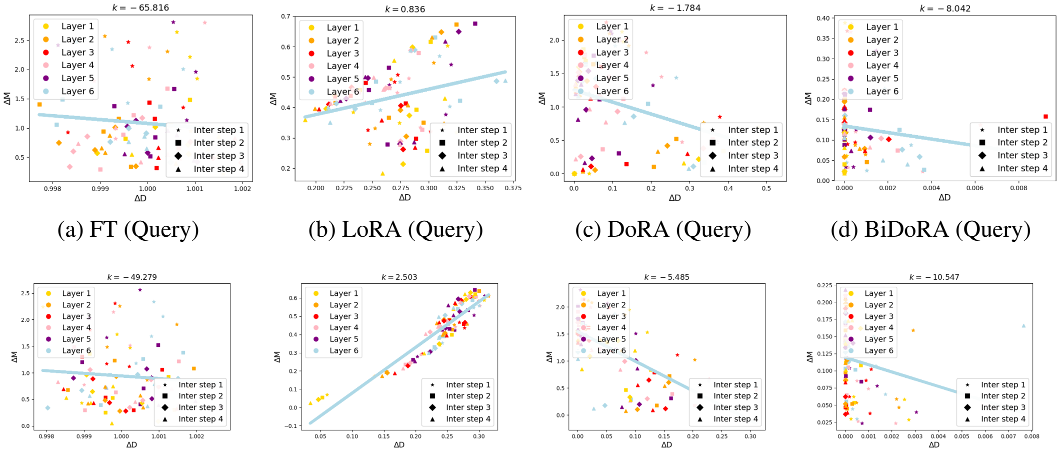
**GPT-2** We summarize the experimental settings for the GPT-2 experiments (Table 7) in Table 9. The experimental configuration, particularly during the inference stage, follows the approach described by Hu et al. (2022b).

## D Weight Decomposition Analysis

Define the weight decomposition of a weight matrix $\mathbf{W} \in \mathbb{R}^{d \times k}$ (e.g., query matrix in an attention layer) as $\mathbf{W} = \mathbf{m}\frac{\mathbf{V}}{\|\mathbf{V}\|_c} = \|\mathbf{W}\|_c \frac{\mathbf{W}}{\|\mathbf{W}\|_c}$, where $\mathbf{m} \in \mathbb{R}^{1 \times k}$ is the magnitude vector, and $\mathbf{V} \in \mathbb{R}^{d \times k}$ is the directional matrix, with $\| \cdot \|_c$ representing the vector-wise norm of a matrix across each

Table 8: The hyperparameters used for RoBERTa on the GLUE benchmark (Wang et al., 2019), Reuters21578 dataset (Padmanabhan et al., 2016), BioNLP dataset (Collier et al., 2004), and CoNLL2003 dataset (Tjong Kim Sang, 2002).

| | Settings | MNLI | SST-2 | MRPC | CoLA | QNLI | QQP | RTE | STS-B | ModApte | ModHayes | ModLewis | BioNLP | CoNLL2003 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Optimizer | AdamW | | | | | | | | | | | | |
| | Warmup Ratio | 0.06 | | | | | | | | | | | | |
| | Scheduler | Linear | | | | | | | | | | | | |
| | LoRA rank | rank = 4 | | | | | | | | | | | | |
| | LoRA $\alpha$ | 8 | | | | | | | | | | | | |
| RoBERTa-base | Total batch size | 32 | | | | | | | | | | | | |
| | Global steps | 20k | 12k | 25k | 20k | 15k | 20k | 15k | 12k | 20k | 20k | 20k | 12k | 12k |
| | Lower learning rate | 5e-5 | 1e-5 | 2e-5 | 5e-5 | 2e-5 | 5e-5 | 1e-5 | 1e-5 | 3e-5 | 3e-5 | 3e-5 | 1e-5 | 2e-5 |
| | Upper learning rate | 5e-5 | 1e-5 | 2e-5 | 5e-5 | 2e-5 | 5e-5 | 1e-5 | 1e-5 | 3e-5 | 3e-5 | 3e-5 | 1e-5 | 2e-5 |
| | Lower weight decay | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 |
| | Upper weight decay | 0.1 | 0.1 | 0.1 | 0.1 | 0 | 0.1 | 0.1 | 0.01 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| | Max Seq Length | 512 | | | | | | | | | | | | |
| | Regularization Coefficient | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 0 | 1e-5 | 0 | 1e-5 | 0 |
| RoBERTa-large | Total batch size | 32 | | | | | | | | | | | | |
| | Global steps | 50k | 20k | 30k | 20k | 60k | 40k | 15k | 10k | 20k | 20k | 20k | 12k | 15k |
| | Lower learning rate | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 2e-5 | 1e-5 |
| | Upper learning rate | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 1e-5 | 2e-5 | 1e-5 |
| | Lower weight decay | 0.5 | 0.5 | 0 | 0.2 | 0.5 | 0.5 | 0.5 | 0.5 | 0.2 | 0.1 | 0.2 | 0.02 | 0.1 |
| | Upper weight decay | 0.5 | 0.05 | 0 | 0.2 | 0.5 | 0.5 | 0.1 | 0.5 | 0.1 | 0.1 | 0.1 | 0.02 | 0.1 |
| | Max Seq Length | 128 | | | | | | | | | | | | |
| | Regularization Coefficient | 0 | 0 | 1e-5 | 1e-5 | 0 | 1e-5 | 0 | 1e-5 | 0 | 1e-5 | 0 | 0 | 1e-5 |



(a) FT (Query) (b) LoRA (Query) (c) DoRA (Query) (d) BiDoRA (Query)

(e) FT (Value) (f) LoRA (Value) (g) DoRA (Value) (h) BiDoRA (Value)

Figure 4: **Magnitude and direction updates for query (top row) and value (bottom row) matrices** for (a) FT, (b) LoRA, (c) DoRA, and (d) BiDoRA across different layers and intermediate steps after fine-tuning the GPT2 model on the E2E dataset (Novikova et al., 2017). Different markers represent matrices from different training steps, while different colors indicate matrices from each layer. The values of correlation are shown in the captions, denoted by $k$.

Table 9: The hyperparameters we used for GPT-2 on the E2E NLG benchmark (Novikova et al., 2017).

| Settings | Training |
|---|---|
| Optimizer | AdamW |
| Warmup Ratio | 0.06 |
| Scheduler | Linear |
| LoRA rank | $\text{rank}_\text{a}$ = $\text{rank}_\text{u}$ = 4 |
| LoRA $\alpha$ | 32 |
| Label Smooth | 0.1 |
| Lower learning rate | 1e-3 |
| Upper learning rate | 1e-4 |
| Lower weight decay | 1 |
| Upper weight decay | 1 |
| Max Seq Length | 512 |
| Regularization Coefficient | 1e-5 |
| **Settings** | **Inference** |
| Beam Size | 10 |
| Length Penalty | 0.9 |
| no repeat ngram size | 4 |

column. This decomposition ensures that each column of $\mathbf{V}/\|\mathbf{V}\|_c$ remains a unit vector, and the corresponding scalar in $\mathbf{m}$ defines the magnitude of each vector. Liu et al. (2024a) examine the magnitude and directional variations between $\mathbf{W_0}$ and $\mathbf{W}_{\mathrm{FT}}$, defined as $\Delta\mathbf{M}_{\mathrm{FT}}^t = \frac{\sum_{n=1}^{k}|\mathbf{m}_{\mathrm{FT}}^{n,t} - \mathbf{m}_0^n|}{k}$ and $\Delta\mathbf{D}_{\mathrm{FT}}^t = \frac{\sum_{n=1}^{k}(1-\cos(\mathbf{V}_{\mathrm{FT}}^{n,t}, \mathbf{W_0}^n))}{k}$. Here, $\Delta\mathbf{M}_{\mathrm{FT}}^t$ and $\Delta\mathbf{D}_{\mathrm{FT}}^t$ represent the magnitude and direction differences between $W_0$ and $W_{\mathrm{FT}}$ at the $t$-th training step, respectively, with $\cos(\cdot,\cdot)$ denoting cosine similarity. $m_{\mathrm{FT}}^{n,t}$ and $m_0^n$ are the $n^{\text{th}}$ scalars in their respective magnitude vectors, while $\mathbf{V}_{\mathrm{FT}}^{n,t}$ and $\mathbf{W_0}^n$ are the $n^{\text{th}}$ columns in $\mathbf{V}_{\mathrm{FT}}^t$ and $\mathbf{W_0}$. Intuitively, a consistent positive slope trend across all the intermediate steps implies a difficulty in concurrent learning of both magnitude and direction, suggesting that slight directional changes are challenging to execute alongside more significant magnitude alterations. In contrast, a relatively negative slope signifies a more varied learning pattern, with a more pronounced negative correlation indicating a larger learning capacity.

One important motivation of DoRA is to bridge the inherent differences between LoRA and FT. Similar to DoRA, we conduct a weight decomposition analysis on the correlation between the change of magnitudes and that of directions for BiDoRA and baseline methods by fine-tuning a GPT2-medium model on the E2E dataset. As shown in Fig. 4, FT, DoRA, and BiDoRA all exhibit negative correlation values, while LoRA shows a positive correlation, consistent with the findings in Liu et al. (2024a). On the query matrix, BiDoRA achieves a negative correlation of $-8.042$, closer to FT than DoRA's $-1.784$. This improvement is attributed to the decoupled training process of the two layers, which allows for a higher learning capacity compared to DoRA. On the value matrix, BiDoRA also achieves a negative correlation of $-10.547$, indicating closer alignment with FT ($-49.279$) compared to DoRA ($-5.485$).

## E Training Cost

Table 10 compares the training efficiency of LoRA, DoRA, and BiDoRA on the GLUE benchmark using the RoBERTa-base model. The table details the total training steps required for convergence and the per-step computational cost, which is normalized relative to LoRA for reference. For a fair comparison, all methods were benchmarked on a single NVIDIA A100 GPU. The results show that BiDoRA converges in fewer steps than LoRA and DoRA, while the per-step cost for BiDoRA is modestly higher, as its BLO process requires iterative updates between the two levels and the computation of hypergradients. The total training time for BiDoRA is approximately 1.64 times that of DoRA, a training cost that remains comparable to the baselines. Given BiDoRA's superior

Table 10: Average training time cost on the GLUE benchmark (Wang et al., 2019).

| Method | LoRA | DoRA | BiDoRA |
|---|---|---|---|
| Per-step cost | ×1 | ×1.36 | ×3.54 |
| Total steps | 27.45k | 27.45k | 17.37k |
| Total time | ×1 | ×1.36 | ×2.24 |

Table 11: Experiment results on different data partitions of BiDoRA.

| Partition | ModApte | ModHayes | ModLewis |
|---|---|---|---|
| 0.6 | 85.32 | 79.76 | 77.69 |
| 0.7 | 85.32 | **80.01** | **77.74** |
| 0.8 | **85.34** | 79.93 | 77.63 |
| 0.9 | 85.27 | 79.85 | 77.64 |
| 1.0 | 85.23 | 79.59 | 77.42 |

performance across various tasks, we argue that this slight increase in computational cost is an acceptable trade-off, underscoring our method's practicality.

## F The Role of Hyperparameter

The hyperparameter tuning for BiDoRA is simple, convenient, and straightforward. We further conducted experiments regarding the dataset partition of $\mathcal{D}_{tr}$ and $\mathcal{D}_{val}$ to provide insights into its role in BiDoRA. The dataset partition helps maintain the balance of inner/outer optimization by assigning different portions of data. The direction component has more trainable parameters, so it is reasonable to use more data for training the lower level while using the remaining data for training magnitudes. As shown in Table 11, we varied the inner-level dataset $\mathcal{D}_{tr}$ partition from $0.6$ to $1.0$ with $0.1$ intervals and experimented with RoBERTa-base on three splits of the Reuters21578 dataset to examine its influence.

The results indicate that both extremes hurt overall performance. When the inner partition is too small ($\leq 0.6$), directions are not well-trained, and when the inner partition is $1.0$, magnitudes are not trained at all, leading to a significant performance drop. These findings demonstrate that BLO is effective in the sense that both levels are necessary for enhancing performance. Although tuning the partition ratio may further improve overall performance, we maintain a consistent data partition of $8:2$ in all the experiments for simplicity. A fixed configuration of data partition already consistently yields superior performance with BiDoRA, demonstrating that our method is robust to this hyperparameter within a certain range.

## G Comparison with Other General Methods for Addressing Overfitting

Common strategies to curb overfitting include stronger weight decay and higher dropout. We evaluate both for DoRA by varying one factor at a time while keeping other hyperparameters at their tuned values. Specifically, we sweep weight decay with fixed dropout, and then sweep dropout with fixed weight decay, using RoBERTa-base across three datasets. Results are shown in Table 12.

Neither higher weight decay nor dropout alone effectively resolves overfitting or improves generalization as much as BiDoRA. In contrast, BiDoRA leverages DoRA's magnitude-direction structure and trains the two components on separate splits, which better regularizes learning. Since BiDoRA does not change the DoRA architecture, it can be combined with these general strategies if desired.

## H Ablation Studies

In this section, we perform ablation studies to investigate the effectiveness of individual modules or strategies in BiDoRA. We fine-tune a RoBERTa-base model on the GLUE benchmark under different ablation settings, and the results are shown in Table 13.

**Retraining.** We test the model directly obtained from the search phase to evaluate the effectiveness of further retraining the direction component. The results show that BiDoRA outperforms BiDoRA (w/o retraining) on average, highlighting the necessity of retraining. Table 13 also validates that retraining the direction component leads to superior performance than retraining the magnitude.

Table 12: Experiment results on different weight decay values and different dropout rates of DoRA.

| Method | CoLA | MRPC | RTE |
|---|---|---|---|
| DoRA (weight decay = 0) | 59.3 | 88.7 | 72.9 |
| DoRA (weight decay = 0.05) | 60.1 | 89.2 | 73.3 |
| DoRA (weight decay = 0.1) | 60.5 | 89.2 | 73.2 |
| DoRA (weight decay = 0.2) | 60.3 | 89.0 | 73.2 |
| DoRA (dropout rate = 0) | 59.2 | 89.2 | 72.9 |
| DoRA (dropout rate = 0.1) | 60.2 | 88.9 | 71.4 |
| DoRA (dropout rate = 0.2) | 55.1 | 87.8 | 64.2 |
| BiDoRA | **61.3** | **89.4** | **76.0** |

Table 13: Ablation studies. We evaluate the performance of BiDoRA without retraining (w/o retraining), without BLO ($\xi = 0$), without orthogonal regularization (w/o cst.), and with retraining magnitude.

| Method | MNLI | SST-2 | MRPC | CoLA | QNLI | QQP | RTE | STS-B | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| BiDoRA (retraining magnitude) | 87.0 | 94.3 | 89.1 | 60.7 | **92.7** | 91.0 | 73.4 | 89.9 | 84.8 |
| BiDoRA (w/o retraining) | 87.0 | 94.2 | 89.0 | 57.3 | 92.4 | 90.6 | 71.6 | 90.0 | 84.0 |
| BiDoRA ($\xi = 0$) | 86.9 | 94.2 | 89.0 | 59.4 | 90.8 | **91.2** | 75.9 | 90.0 | 84.7 |
| BiDoRA (w/o cst.) | 87.0 | **94.4** | 88.6 | **61.3** | **92.7** | 90.2 | 76.0 | **90.1** | 85.0 |
| BiDoRA | **87.1** | **94.4** | **89.4** | **61.3** | **92.7** | 90.6 | **76.1** | **90.1** | **85.2** |

**Bi-level optimization.** We set $\xi$ to zero in Algorithm 1 to assess the effectiveness of the BLO framework. This ablation setting can be interpreted as an alternative learning method where two optimization steps are carried out alternately on two different splits of the training dataset. Notably, in the alternative learning method, the updating of each component is unaware of the others, making the training less stable. In contrast, the hyper-gradient used in BLO avoids this issue by connecting the two levels in a certain way. The results show that BiDoRA outperforms BiDoRA ($\xi = 0$) on average, demonstrating the efficacy of the BLO strategy.

**Orthogonal regularization.** We examine the effectiveness of the orthogonality constraint in Eq. (1) by setting $\gamma$ to zero. Results show that BiDoRA outperforms BiDoRA (w/o cst.) on average, indicating the effectiveness of applying the orthogonality regularizer to alleviate overfitting.

# I Discussion

The advantage of BiDoRA is supported by both theoretical insights and empirical evidence, as detailed as follows. We also discuss the limitations of BiDoRA.

**Motivation.** Theoretically, Liu et al. (2024a) showed that LoRA's training pattern tends to be coupled in terms of magnitude-direction correlation, which degrades learning capacity. Their solution was to introduce a reparameterization that decouples these components in the formulation. We build upon DoRA following their theory and further decouple magnitude and direction in terms of training dynamics. Specifically, the two components are trained in separate loops within a bilevel optimization framework, which is expected to improve performance in an intuition similar to DoRA.

Besides, a similar strategy of combating overfitting based on BLO has been utilized in the well-established practice of differentiable neural architecture search (DARTS, Liu et al. (2019)), where architecture and subnetworks are learned using different dataset splits. Optimizing the selection variables and subnetworks in a single loop can result in an over-expressive network since the selection variables tend to select all subnetworks to achieve the best expressiveness, which, however, incurs severe overfitting. In contrast, training the subnetworks with the selection module fixed on the training split while validating the effectiveness of the selection module on the unseen validation split effectively eliminates the risk of overfitting. Similarly, we treat the **magnitude component as the architecture** and the **direction component as the subnetworks** and train these components on **separate datasets**. As shown in Table 5, BiDoRA demonstrates better resistance to overfitting compared to DoRA, given the smaller performance gap between the training set and test set. Furthermore, the asynchronous gradient update steps at the two optimization levels in BiDoRA facilitate better decoupling of the two components, leading to a more flexible update pattern that closely resembles FT. As illustrated in Fig. 4, the updates across different layers using BiDoRA have a correlation value that is closest to that of FT, highlighting its superior learning capability compared to both DoRA and LoRA.

While this work focuses on the empirical validation of BiDoRA, our choice of optimization strategy is grounded in established theoretical research. The convergence properties of similar gradient-based bi-level algorithms have been previously analyzed (Pedregosa, 2016; Rajeswaran et al., 2019), providing confidence in the stability of our training procedure. Furthermore, the ability of such frameworks to improve generalization—a core objective of BiDoRA—has also been formally studied (Bao et al., 2021), supporting the rationale that our approach can mitigate overfitting.

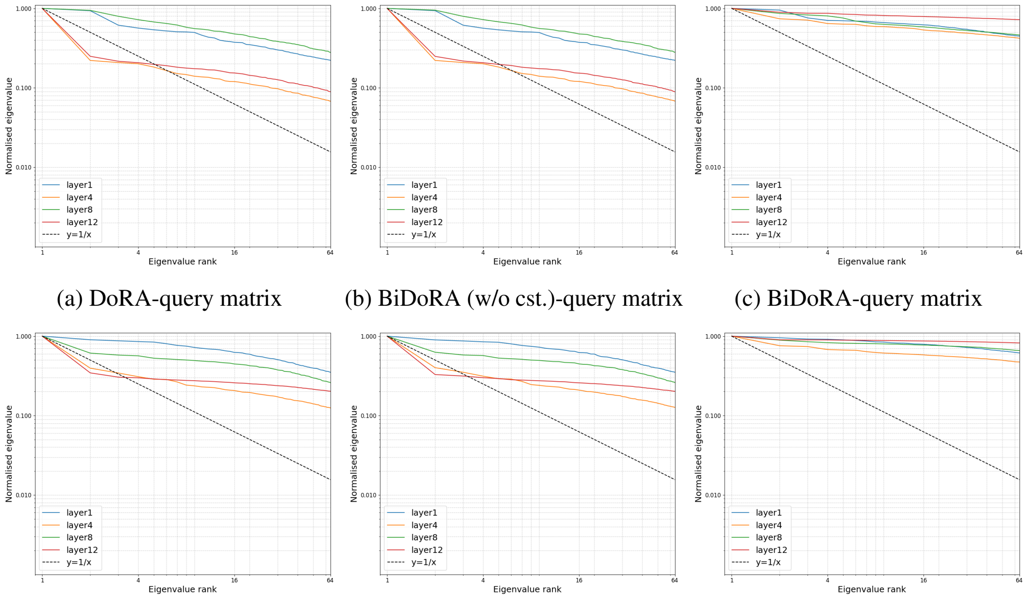
**Empirical evidence.** We performed a Wilcoxon signed-rank test to compare the performance of DoRA and BiDoRA. Specifically, we used the results from Table 3. For each PEFT method, we collected 9 values (8 values from each dataset plus the average performance) from one base model. We concatenated the results from three base models (RoBERTa-base, RoBERTa-large, and DeBERTa-XXL) to obtain a list of 27 values. A comparison of these 27 values between DoRA and BiDoRA reveals that BiDoRA is significantly better than DoRA, with a p-value of $2.4 \times 10^{-4}$. This result demonstrates that BiDoRA offers a non-marginal improvement over DoRA.

Additionally, the weight decomposition analysis (Fig. 4), indicates that BiDoRA achieves better decoupling of the components compared to DoRA. Evaluation metrics across various tasks demonstrate the superior performance of BiDoRA, confirming that our decoupled optimization loop leads to improved outcomes.

**Limitations.** One potential limitation of BiDoRA is its training efficiency (see Section E) in terms of per-step cost, which could be reduced by using more advanced hyper-gradient estimators, such as SAMA (Choe et al., 2023a) or MixFlow-MG (Kemaev et al., 2025). Furthermore, while we have empirically shown that BiDoRA induces better decoupling between the magnitude and direction components (Fig. 4), a formal theoretical analysis of this property is currently lacking and serves for future work.

## J Evidence on Orthogonality of Incremental Matrix

To verify that the orthogonal regularization (OR) proposed in Section 3 encourages the columns of the direction matrix to be orthogonal, we visualize the normalized eigenvalues of the matrix in Fig. 5. The results show that, compared to methods without OR (i.e., DoRA and BiDoRA w/o cst.), BiDoRA with OR produces eigenvalues that are more closely aligned with those of a purely orthogonal matrix, where all eigenvalues would be one. This effect holds for both the query and value matrices and verifies the effectiveness of the OR constraint.



(a) DoRA-query matrix (b) BiDoRA (w/o cst.)-query matrix (c) BiDoRA-query matrix

(d) DoRA-value matrix (e) BiDoRA (w/o cst.)-value matrix (f) BiDoRA-value matrix

Figure 5: Eigenspectra of the direction matrix for query (top) and value (bottom) matrices across different layers

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state our contributions and scope (BiDoRA's BLO decoupling, overfitting reduction, and performance). Evidence appears in Sections C and 4, with ablations in Table 13, gap analysis in Table 5, weight-decomposition analysis in Section D and Fig. 4, and training cost in Section E.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss limitations in Section I (training efficiency and lack of formal theory) and quantify compute in Section E. We also note avenues for future theoretical analysis.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.

- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not present formal theorems or proofs. It provides algorithmic details and notation (Algorithm 1 and Eqs. (1) and (4)) but no theoretical results requiring assumptions or proofs.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Datasets and models are described in Section B; experimental settings and hyperparameters in Section C.4; baselines in Section B.2. The BLO split rule and stopping criteria are in Sections C and E, enabling reproduction of the main claims.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We include an anonymized implementation and instructions in the supplemental material sufficient to reproduce the main experimental results (see Sections B.2, C and C.4). Public datasets are cited and accessible; a public repository will be provided after acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify data splits (e.g., 8:2 train/validation for BLO in Section C), hyperparameters and training details in Section C.4, datasets in Section B, and optimizers/libraries in Section C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report a Wilcoxon signed-rank test comparing BiDoRA and DoRA (p = $2.4 \times 10^{-4}$) in Section I. We also summarize results with mean and standard deviation where applicable (e.g., Fig. 2).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: Section E specifies the hardware (single NVIDIA A100 GPU) and compares per-step cost, total steps, and total time across methods, allowing estimation of compute requirements.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We use public datasets, do not process personal or sensitive data, and adhere to standard reproducibility and citation practices. No human subjects are involved and no privacy-sensitive information is used.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper focuses on a foundational fine-tuning method and does not include a separate broader-impact discussion. Potential risks include easier adaptation of models that may inherit dataset biases; benefits include improved efficiency and accuracy. We can add a brief discussion at camera-ready if preferred.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release high-risk datasets or models; we evaluate on existing public models and datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Our implementation is released under the Apache License 2.0. Third-party libraries (e.g., Transformers) are used under their respective licenses (Apache-2.0 for Transformers), and datasets are used under their original terms with citations provided. We include a LICENSE file and license notices in the supplemental material.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or human subjects.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or human subjects; IRB approval is not required.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The core method does not use LLMs as a tool in development; LMs are the objects of evaluation. No declaration is required under the NeurIPS LLM policy.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

Justification: We used LLMs only for writing and editing the manuscript; this did not impact the methodology, experiments, or originality of the research. Per policy, no declaration of method usage is required.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.