IPA: An Information-Preserving Input Projection Framework for Model Adaptation

Yuan Yin[1] **Shashanka Venkataramanan**[1] **Tuan-Hung Vu**[1]
Andrei Bursuc[1] **Matthieu Cord**[1,2]
[1]Valeo.ai, Paris, France [2]Sorbonne Université, CNRS, ISIR, F-75005 Paris, France
{firstname.lastname}@valeo.com

Abstract

Parameter-efficient fine-tuning (PEFT) methods, such as LoRA, reduce adaptation cost by injecting low-rank updates into pretrained weights. However, LoRA's down-projection is randomly initialized and data-agnostic, discarding potentially useful information. Prior analyses show that this projection changes little during training, while the up-projection carries most of the adaptation, making the random input compression a performance bottleneck. We propose IPA, a feature-aware projection framework that explicitly preserves information in the reduced hidden space. In the linear case, we instantiate IPA with algorithms approximating top principal components, enabling efficient projector pretraining with negligible inference overhead. Across language and vision benchmarks, IPA improves over LoRA and DoRA, achieving on average 1.5 points higher accuracy on commonsense reasoning and 2.3 points on VTAB-1k, while matching best baseline performance with roughly half the trainable parameters when the projection is frozen.

1 Introduction

Adapting large foundation models is challenging since full fine-tuning is costly (Houlsby et al., 2019; Hu et al., 2022). To address this bottleneck, the community has developed a range of parameter-efficient fine-tuning (PEFT) methods that reduce the number of trainable parameters by an order of magnitude compared to the base model (see surveys, e.g., Han et al., 2024; Zhang et al., 2025). Among these, Low-Rank Adaptation (LoRA; Hu et al., 2022) has gained significant traction due to its simplicity and effectiveness in the large-language-model community. In LoRA, each target weight matrix is reparameterized as the sum of the original pre-trained weight W and a low-rank update $\Delta W = BA$, where A (the "down" projection) maps inputs into a lower-dimensional space and B (the "up" projection) maps them back.

Although there has been a flurry of follow-up works to LoRA, most focus on alternative initializations (Meng et al., 2024; Yang et al., 2024) or extended structures (Liu et al., 2024; Huang et al., 2025; Albert et al., 2025) by restricting their analysis to the pretrained weight matrix, while paying little attention to the distribution of input features. In contrast, we broaden the focus to explicitly account for the role of input features. In the original LoRA formulation, the down-projection matrix A is randomly initialized and thus data-agnostic. Analyses of LoRA's inherent asymmetry show that during adaptation, this down-projection A remains close to its initialization, whereas the up-projection B adapts more effectively to the data (Tian et al., 2024; Hayou et al., 2024b). This suggests that a data-agnostic input projection can become a performance bottleneck, motivating its replacement with a feature-aware, data-dependent alternative that better aligns with the intrinsic structure of the inputs.

In this paper, we pursue this direction and introduce IPA, an input-feature-aware projection scheme designed to preserve information in the adapter's hidden feature space. Our contributions are: • We

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Continual and Compatible Foundation Model Updates (CCFM).

formulate adaptation with a dedicated feature-projection pretraining objective that maximizes information preservation in the bottleneck dimension through an encoder–decoder formulation. • We instantiate this framework in the linear setting using efficient forward-only pretraining algorithms. • We empirically validate IPA on language and vision-language tasks, showing consistent improvements over random linear projections. On several architectures, IPA matches the performance of fully trained LoRA while requiring roughly half as many trainable parameters.

2 IPA: Information Preserving Input Projection for Adaptation

2.1 Preliminaries: LoRA

Given a pretrained weight $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ defining $f_W(x) = Wx$, LoRA augments it with two low-rank maps: $f_A \colon x \mapsto Ax \in \mathbb{R}^r$, $f_B \colon x_h \mapsto Bx_h \in \mathbb{R}^{d_{\text{out}}}$, where $A \in \mathbb{R}^{r \times d_{\text{in}}}$, $B \in \mathbb{R}^{d_{\text{out}} \times r}$, and $r \ll \min(d_{\text{in}}, d_{\text{out}})$. At step t, the adapted forward pass is

$$z = f_W(x) + \lambda f_{B_t}(f_{A_t}(x)) = Wx + \lambda B_t A_t x, \tag{1}$$

The elements of A_0 are drawn from a zero-mean Gaussian (or uniform) distribution and $B_0 = 0$. The positive scalar λ rescales the low-rank residual update. In the original LoRA formulation, $\lambda = \frac{\alpha}{r}$ with $\alpha > 0$. Training LoRA thus implies computing gradients only for A_t and B_t, leaving W unchanged.

2.2 Asymmetric Behaviors in LoRA

While LoRA has been widely adopted for efficient fine-tuning of large pretrained models, we observe a notable asymmetry between its two projection matrices: the down-projection matrix A primarily serves to compress input features into a low-dimensional subspace, whereas the up-projection matrix B plays the critical role of recombining those features to adapt the final model outputs. Notably, tuning B alone while keeping A fixed and randomly initialized often yields performance comparable to tuning both. This suggests that B is mainly responsible for adapting the output, whereas A serves as a feature projector. We provide an empirical analysis in Appendix B.

Implications. These observations indicate that the down-projection matrix A in standard LoRA operates primarily as a random feature projector, rather than encodes the task-specific distinctions. Recent studies of LoRA (Hayou et al., 2024b; Tian et al., 2024) arrive at similar conclusions, showing that standard LoRA induces pronounced asymmetries in both learning dynamics and representational behavior. Consequently, replacing this data-agnostic projector with a more expressive, task-aware map could yield richer hidden representations and improve adaptation performance.

2.3 The IPA Framework

We reinterpret the adaptation scheme by introducing a general function $\mathcal{P}$ and write

$$z = f_W(x) + \lambda f_{B_t}(\mathcal{P}(x)) = Wx + \lambda B_t \mathcal{P}(x),$$

where $\mathcal{P} \colon \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_h}$ projects the input x into a hidden feature $x_h = \mathcal{P}(x) \in \mathbb{R}^{d_h}$ and $B_0 = 0$.

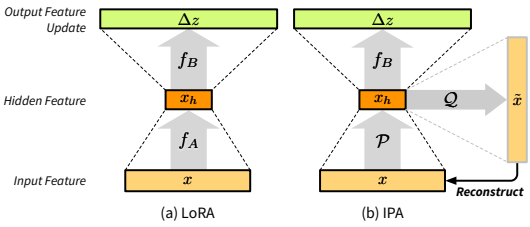


Figure 1: **IPA vs. LoRA.** IPA adds pretrained projectors $\mathcal{P}, \mathcal{Q}$ for reconstruction, keeping only $\mathcal{P}$ during adaptation.

Information preserving input projection. When $d_h < d_{\text{in}}$, the projection $\mathcal{P}$ must compress x, which risks discarding task-relevant information. Standard LoRA initializes $\mathcal{P}$ as a random linear map, thus ignoring the input distribution. To address this, we instead seek $\mathcal{P}$ (and a complementary decoder $\mathcal{Q} \colon \mathbb{R}^{d_h} \to \mathbb{R}^{d_{\text{in}}}$) that minimize the reconstruction error:

$$\min_{\mathcal{P},\mathcal{Q}} \mathbb{E}_{x \sim \mathrm{p}(x)} \|x - \tilde{x}\|^2, \quad \text{where} \quad \tilde{x} = \mathcal{Q}(\mathcal{P}(x)). \tag{2}$$

This objective encourages $\mathcal{P}$ to preserve as much information from the original input as possible, as measured by the L^2 reconstruction loss. Fig. 1 contrasts IPA with LoRA.

Forward-only pretraining of projector. Eq. (2) corresponds precisely to the objective of an autoencoder. One could therefore imagine training it with either linear or nonlinear functions for $\mathcal{P}$

and $\mathcal{Q}$. However, doing so for each modulated layer via backpropagation is impractical: the loss is difficult to integrate into the adapter training pipeline and incurs significant computational overhead compared to LoRA. Instead, we propose to learn the projector in a *forward-only* manner.

2.4 Instantiation: Linear Case

To instantiate the framework in practice, we must specify (i) the distribution of features used to pretrain the projector $\mathcal{P}$, (ii) the form of the projector $\mathcal{P}$, and (iii) the algorithm used to pretrain it.

Pretraining distribution. We pretrain $\mathcal{P}$ using target-domain hidden representations. Concretely, we pass training tokens through the frozen pretrained model and collect the resulting layer-wise intermediate features, forming a pretraining set $\hat{X} = [\hat{x}_i]_{i=1}^{N} \in \mathbb{R}^{N \times d_{\text{in}}}$. This ensures that $\hat{X}$ reflects both the model's internal feature and the target-domain data distributions.

Projector architecture. To preserve LoRA's inference-time efficiency, we restrict $\mathcal{P}$ and its decoder $\mathcal{Q}$ to linear maps defined by a shared matrix $U \in \mathbb{R}^{d_h \times d_{\text{in}}}$: $\mathcal{P}(x) = Ux, \mathcal{Q}(x_h) = U^\top x_h$. Solving Eq. (2) then reduces to computing the top-d_h eigenvectors of the empirical covariance $\Sigma = \frac{1}{N}\hat{X}^\top \hat{X}$.

Pretraining algorithm. Full PCA over all hidden states is infeasible due to storage and compute costs. Instead, we adopt incremental PCA (IPCA; Ross et al., 2008), which processes feature mini-batches sequentially and updates a low-rank approximation of Σ. Alternatives such as the generalized Hebbian algorithm (GHA; Sanger, 1989) also approximate principal components, but we found IPCA both more efficient and slightly more accurate in practice (see Appendix C.2).

Default Configuration. Unless otherwise specified, we use target-domain hidden representations as input, a linear projector, and IPCA for pretraining. All main experiments adopt IPCA, and IPA refers to this implementation unless noted otherwise. The projector U can optionally be refined by backpropagating the task loss. We analyze the effect of projector fine-tuning in Section 3.

3 Experiments

3.1 Experimental Setting

Language tasks. We follow the instruction-following protocol of Hu et al. (2023) on the `commonsense-170k` dataset, adapting four LLMs (LLAMA-2 7B (Touvron et al., 2023), LLAMA-3 8B (Grattafiori et al., 2024), QWEN-2.5 7B (Qwen et al., 2024), GEMMA-3 4B (Gemma Team et al., 2025)) for 3 epochs and evaluating on their test splits.

Vision tasks. For open-vocabulary classification we use VTAB-1k (Zhai et al., 2019), grouped into *Natural*, *Specialized*, and *Structured*, with 1000 examples per task. We adapt the SIGLIP-2 backbone (Tschannen et al., 2025) by tuning only the vision encoder with cross-entropy on image-text similarity scores. Evaluation follows Zhang et al. (2022), we report the best test accuracy over 100 epochs.

Baselines. We compare IPA with: (i) **LoRA** (Hu et al., 2022), low-rank adapters with random down-projection; (ii) **DoRA** (Liu et al., 2024), which decomposes weights into magnitude and direction, applying LoRA to the latter. Both have fixed (✗) and trainable (✓) projector variants.

Hyperparameters. We use Adam (Kingma & Ba, 2015) with linear warm-up, fixing all settings except base learning rate (aligned with Liu et al. (2024) for LLAMA-2/3; tuned for newer models). Details are in Appendix C.1. All methods use the same adapter dimensions ($d_h = 32$ for language, $d_h = 8$ for vision), ensuring differences stem only from projector training. For projector pretraining, IPA uses 10% of commonsense-170k and the full VTAB-1k sets (see Appendix C.2).

3.2 Main Results

IPA improves adaptation over random projection. Tables 1 and 2 summarize our accuracy results on the instruction-following benchmark and the open-vocabulary classification tasks, respectively. On the instruction-following benchmark, at hidden dimension $d_h = 32$ IPA outperforms both LoRA and DoRA across most configurations and base models. For example, in Table 1, on LLAMA-3 8B without projector fine-tuning, IPA achieves an average accuracy of 85.6%, outperforming LoRA (85.0%) by 0.6 points and DoRA (84.7%) by 0.9 points. Even with projector fine-tuning, IPA still leads with 85.9%, compared to 85.5% for LoRA and 85.1% for DoRA. Similar gains are observed

Table 1: **Comparison of instruction-following answer accuracy (%) on commonsense reasoning benchmarks.** All methods are compared in the configuration with (✓) and without (✗) projector finetuning. We highlight the **best** and the second scores under the same projector finetuning setting.

Base model	Method	Proj. FT	Trainable Params (%)	BoolQ	PIQA	SocialIQA	HellaSwag	WinoGrande	ARC-easy	ARC-challenge	OpenbookQA	Avg.
LLAMA-2 7B	LoRA	✗	28.0M (0.41%)	60.5	78.7	74.5	76.3	75.1	82.8	66.1	76.8	73.8
	DoRA	✗	28.9M (0.43%)	58.0	82.0	33.5	12.8	42.1	64.9	43.9	68.4	50.7
	IPA (Ours)	✗	28.0M (0.41%)	**71.7**	**83.2**	**80.0**	**89.0**	**82.0**	**84.8**	**70.1**	**79.0**	**80.0**
	LoRA	✓	56.1M (0.83%)	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
	DoRA	✓	57.0M (0.84%)	**71.8**	83.7	76.0	89.1	82.6	83.7	68.2	**82.4**	79.7
	IPA (Ours)	✓	56.1M (0.83%)	71.1	**84.4**	**80.9**	**90.5**	**82.7**	**85.6**	**71.5**	81.4	**81.1**
LLAMA-3 8B	LoRA	✗	25.2M (0.31%)	73.6	88.1	80.3	95.0	85.2	90.4	**80.1**	87.4	85.0
	DoRA	✗	26.0M (0.32%)	74.3	87.9	79.7	95.3	84.2	90.3	79.5	86.2	84.7
	IPA (Ours)	✗	25.2M (0.31%)	**74.8**	**88.6**	**81.1**	**95.4**	**85.6**	**91.7**	79.9	**87.8**	**85.6**
	LoRA	✓	56.6M (0.70%)	**75.4**	88.6	80.7	95.4	**86.2**	**91.2**	**80.1**	86.1	85.5
	DoRA	✓	57.4M (0.71%)	75.3	89.3	80.8	95.3	85.8	89.9	79.3	85.6	85.1
	IPA (Ours)	✓	56.6M (0.70%)	75.0	**89.9**	**81.2**	**96.0**	85.9	**91.2**	79.6	**88.4**	**85.9**
QWEN-2.5 7B	LoRA	✗	24.3M (0.32%)	62.8	89.3	79.9	94.6	83.1	95.9	88.6	91.4	85.7
	DoRA	✗	25.1M (0.33%)	62.0	89.8	78.6	94.6	83.0	**96.1**	**88.9**	89.8	85.3
	IPA (Ours)	✗	24.3M (0.32%)	**73.3**	**90.0**	**80.2**	**95.0**	**85.2**	95.8	88.8	**92.4**	**87.6**
	LoRA	✓	54.1M (0.71%)	63.5	89.8	79.5	**95.4**	**85.9**	95.9	88.3	**92.2**	86.3
	DoRA	✓	54.9M (0.72%)	**74.5**	**90.0**	**80.2**	**95.4**	**85.9**	95.7	87.7	91.8	87.6
	IPA (Ours)	✓	54.1M (0.71%)	**74.5**	**90.0**	79.7	95.3	85.5	**96.2**	**88.7**	92.0	**87.7**
GEMMA-3 4B	LoRA	✗	21.4M (0.49%)	**69.3**	84.4	78.2	90.6	80.3	89.5	76.4	82.0	81.3
	DoRA	✗	22.0M (0.51%)	69.1	84.2	77.9	**91.0**	80.5	89.4	**78.1**	82.2	81.5
	IPA (Ours)	✗	21.4M (0.49%)	68.7	**85.0**	**78.5**	90.0	**81.5**	**90.3**	78.0	**84.4**	**82.0**
	LoRA	✓	46.6M (1.07%)	70.3	86.0	79.7	93.1	82.3	89.7	79.7	84.4	83.1
	DoRA	✓	47.3M (1.09%)	**70.6**	85.3	**80.0**	92.9	82.8	90.0	77.6	85.4	83.1
	IPA (Ours)	✓	46.6M (1.07%)	69.8	**86.3**	78.8	**93.4**	**83.3**	**90.7**	**80.3**	**86.0**	**83.6**

Table 2: **Accuracy of vision encoder adaptation on VTAB-1k with the SIGLIP-2 base model.** "Vision QV FT" tunes the query/value projections, while "Vision Full FT" tunes the whole vision encoder. We highlight the **best** and second scores under the same setting. We report per-group averages, the *Macro Avg.* (mean of group averages), and the *Micro Avg.* (mean over all tasks).

			Group 1: *Natural*								Group 2: *Specialized*					Group 3: *Structured*											
Method	Proj. FT	Trainable Params (%)	Caltech101	CIFAR-100	DTD	Flowers102	Pets	Sun397	SVHN	G1 Avg.	Camelyon	EuroSAT	Resisc45	Retinopathy	G2 Avg.	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Elev	G3 Avg.	Macro Avg.	Micro Avg.
Zero-shot	—	0 (0.00%)	84.4	73.9	63.0	84.1	94.9	61.2	28.6	70.0	50.9	40.0	62.8	5.0	39.7	27.9	20.0	17.0	4.2	6.4	4.8	5.2	10.4	12.0	40.5	36.7
Vision QV FT	—	14.2M (3.8%)	94.2	78.3	80.2	98.2	93.3	66.6	93.2	85.2	85.3	96.5	91.0	74.8	86.9	85.0	60.2	48.5	85.1	88.2	52.2	37.2	43.6	62.5	78.2	75.5
Vision Full FT	—	92.9M (24.8%)	94.8	81.5	81.3	98.3	94.7	67.7	93.2	86.1	85.0	96.3	91.5	75.1	87.0	84.1	60.8	42.8	86.1	51.5	83.0	29.7	41.8	60.0	77.7	74.7
LoRA	✗	0.15M (0.039%)	89.0	81.8	75.4	94.3	95.3	64.6	89.9	84.3	79.2	**95.8**	87.0	72.5	83.7	**85.0**	52.2	26.8	71.4	65.9	17.5	8.8	24.0	43.9	70.7	66.0
DoRA	✗	0.17M (0.044%)	89.6	**82.0**	76.0	94.5	**95.4**	64.8	90.3	84.7	79.3	95.7	86.8	72.5	83.6	84.8	54.5	28.3	67.2	68.0	17.6	9.1	25.8	44.4	70.9	66.3
IPA (Ours)	✗	0.15M (0.039%)	**93.1**	81.7	**77.7**	**95.3**	95.1	**65.2**	**90.7**	**85.5**	**81.5**	95.7	**87.3**	**73.3**	**84.5**	83.5	**59.7**	**29.2**	**81.4**	**75.0**	**25.1**	**15.8**	**38.6**	**51.0**	**73.7**	**69.5**
LoRA	✓	0.29M (0.079%)	**94.8**	80.8	75.4	95.8	**95.2**	**65.6**	91.4	85.9	82.4	96.1	88.0	74.0	85.1	**91.8**	58.5	34.7	83.1	76.8	38.4	18.2	38.0	54.9	75.3	71.5
DoRA	✓	0.33M (0.083%)	94.5	81.1	78.1	95.8	**95.2**	**65.6**	91.4	85.7	**83.5**	96.0	87.6	74.1	85.3	91.5	60.6	35.3	**84.5**	78.4	35.3	17.0	37.1	55.0	75.3	71.5
IPA (Ours)	✓	0.29M (0.079%)	**94.8**	**81.3**	**79.8**	**96.3**	94.7	**65.6**	**91.8**	**86.3**	83.0	**96.5**	**88.5**	**74.4**	**85.6**	90.0	**62.5**	**39.5**	82.1	**79.5**	**40.8**	**22.3**	**44.3**	**57.6**	**76.5**	**72.9**

across other base models, yielding an average gain of 1.5 points. On the VTAB-1k benchmark (Table 2), at $d_h = 8$, IPA reaches 73.7% group-level macro average accuracy without projector fine-tuning, surpassing LoRA by 3.0 points and DoRA by 2.8 points. With projector fine-tuning, performance improves to 76.5%, a 1.8-point gain over both baselines.

IPA suffers less from fixing input projectors. As shown in Fig. 2, IPA degrades much less when projectors are fixed: on LLAMA-2 7B the drop is 1.1 points (vs. 3.8 for LoRA and 29.0 for DoRA), and on QWEN-2.5 7B only 0.1 (vs. 0.6 and 2.3). Moreover, IPA without projector finetuning matches or exceeds the best finetuned baseline in 3 of 4 models while using over 50% fewer parameters, surpassing both baselines on LLAMA-2 7B, and slightly outperforming them on LLAMA-3 8B.

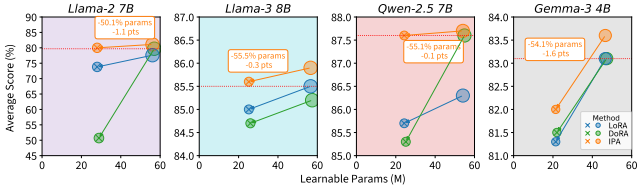


Figure 2: Comparison of IPA and baselines on the commonsense benchmark, with (◯) and without (⊗) projector finetuning. The red dashed line marks the best baseline.

4 Conclusion

We introduced IPA, a framework for parameter-efficient adaptation that replaces random input projection with an information-preserving one. Using a simple batched PCA pretraining, IPA learns meaningful projections without backpropagation. Across language and vision benchmarks, IPA consistently outperforms PEFT baselines with minimal extra cost, showing that data-driven projections enable more expressive and adaptable models.

Acknowledgment

This work was granted access to the HPC resources of IDRIS under the allocation 2024-AD011015854 made by GENCI.

References

Paul Albert, Frederic Z. Zhang, Hemanth Saratchandran, Cristian Rodriguez Opazo, Anton van den Hengel, and Ehsan Abbasnejad. RandLoRA: Full rank parameter-efficient fine-tuning of large models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=Hn5eoTunHN. 1, 9

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen et al. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25:70:1–70:53, 2024. URL https://jmlr.org/papers/v25/23-0870.html. 9

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-bastien Grill, Sabela Ramos et al. Gemma 3 Technical Report. *arXiv e-prints*, art. arXiv:2503.19786, March 2025. doi: 10.48550/arXiv.2503.19786. 3

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang et al. The Llama 3 Herd of Models. *arXiv e-prints*, art. arXiv:2407.21783, July 2024. doi: 10.48550/arXiv.2407.21783. 3

Demi Guo, Alexander M. Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 4884–4896. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.378. URL https://doi.org/10.18653/v1/2021.acl-long.378. 9

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=lIsCS8b6zj. 1, 9

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. The impact of initialization on lora finetuning dynamics. 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*, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/d4387c37b3b06e55f86eccdb8cd1f829-Abstract-Conference.html. 9

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=NEv8YqBROO. 1, 2, 9

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, 2022a. URL https://openreview.net/forum?id=0RDcd5Axok. 9

Shwai He, Liang Ding, Daize Dong, Jeremy Zhang, and Dacheng Tao. SparseAdapter: An easy approach for improving the parameter-efficiency of adapters. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 2184–2190. Association for Computational Linguistics, 2022b. doi: 10.18653/V1/2022.FINDINGS-EMNLP.160. URL https://doi.org/10.18653/v1/2022.findings-emnlp.160. 9

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. 1, 9

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, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9. 1, 3, 9

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. LLM-Adapters: An adapter family for parameter-efficient fine-tuning of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 5254–5276. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.319. URL https://doi.org/10.18653/v1/2023.emnlp-main.319. 3, 10

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. LoraHub: Efficient cross-task generalization via dynamic LoRA composition. *CoRR*, abs/2307.13269, 2023. doi: 10.48550/ARXIV.2307.13269. URL https://doi.org/10.48550/arXiv.2307.13269. 9

Qiushi Huang, Tom Ko, Zhan Zhuang, Lilian Tang, and Yu Zhang. Hira: Parameter-efficient hadamard high-rank adaptation for large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=TwJrTz9cRS. 1, 9

Shibo Jie, Haoqing Wang, and Zhi-Hong Deng. Revisiting the parameter efficiency of adapters from the perspective of precision redundancy. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 17171–17180. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01579. URL https://doi.org/10.1109/ICCV51070.2023.01579. 9

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980. 3

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. 9

Minglei Li, Peng Ye, Yongqi Huang, Lin Zhang, Tao Chen, Tong He, Jiayuan Fan, and Wanli Ouyang. Adapter-X: A novel general parameter-efficient fine-tuning framework for vision. *CoRR*, abs/2406.03051, 2024. doi: 10.48550/ARXIV.2406.03051. URL https://doi.org/10.48550/arXiv.2406.03051. 9

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, 2024. URL https://openreview.net/forum?id=3d5CIRG1n2. 1, 3, 9, 10

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 565–576. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.47. URL https://doi.org/10.18653/v1/2021.acl-long.47. 9

Yuren Mao, Yuhang Ge, Yijiang Fan, Wenyi Xu, Yu Mi, Zhonghao Hu, and Yunjun Gao. A survey on LoRA of large language models. *Frontiers Comput. Sci.*, 19(7):197605, 2025. doi: 10.1007/S11704-024-40663-9. URL https://doi.org/10.1007/s11704-024-40663-9. 9

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. 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/db36f4d603cc9e3a2a5e10b93e6428f2-Abstract-Conference.html. 1, 9

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang et al. Qwen2.5 Technical Report. *arXiv e-prints*, art. arXiv:2412.15115, December 2024. doi: 10.48550/arXiv.2412.15115. 3

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 506–516, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/e7b24b112a44fdd9ee93bdf998c6ca0e-Abstract.html. 9

David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *Int. J. Comput. Vis.*, 77(1-3):125–141, 2008. doi: 10.1007/S11263-007-0075-7. URL https://doi.org/10.1007/s11263-007-0075-7. 3

Terence D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2(6):459–473, 1989. doi: 10.1016/0893-6080(89)90044-0. URL https://doi.org/10.1016/0893-6080(89)90044-0. 3

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. Lora vs full fine-tuning: An illusion of equivalence. *CoRR*, abs/2410.21228, 2024. doi: 10.48550/ARXIV.2410.21228. URL https://doi.org/10.48550/arXiv.2410.21228. 9

Yi-Lin Sung, Varun Nair, and Colin Raffel. Training neural networks with fixed sparse masks. 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. 24193–24205, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/cb2653f548f8709598e8b5156738cc51-Abstract.html. 9

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 13003–13051. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.824. URL https://doi.org/10.18653/v1/2023.findings-acl.824. 9

Pengwei Tang, Yong Liu, Dongjie Zhang, Xing Wu, and Debing Zhang. LoRA-Null: Low-rank adaptation via null space for large language models. *CoRR*, abs/2503.02659, 2025. doi: 10.48550/ARXIV.2503.02659. URL https://doi.org/10.48550/arXiv.2503.02659. 9

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng-Zhong Xu. HydraLoRA: An asymmetric LoRA architecture for efficient fine-tuning. 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/123fd8a56501194823c8e0dca00733df-Abstract-Conference.html. 1, 2, 9

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull et al. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288. 3

Michael Tschannen, Alexey A. Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier J. Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *CoRR*, abs/2502.14786, 2025. doi: 10.48550/ARXIV.2502.14786. URL https://doi.org/10.48550/arXiv.2502.14786. 3

Xujia Wang, Haiyan Zhao, Shuo Wang, Hanqing Wang, and Zhiyuan Liu. MALoRA: Mixture of asymmetric low-rank adaptation for enhanced multi-task learning. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pp. 5609–5626. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-NAACL.312. URL https://doi.org/10.18653/v1/2025.findings-naacl.312. 9

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. Corda: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. 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/83f95bb0ac5046338ea2afe3390e9f4b-Abstract-Conference.html. 1, 9

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, André Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet et al. The visual task adaptation benchmark. *CoRR*, abs/1910.04867, 2019. URL http://arxiv.org/abs/1910.04867. 3

Dan Zhang, Tao Feng, Lilong Xue, Yuandong Wang, Yuxiao Dong, and Jie Tang. Parameter-Efficient Fine-Tuning for Foundation Models. *arXiv e-prints*, art. arXiv:2501.13787, January 2025. doi: 10.48550/arXiv.2501.13787. 1

Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. Neural prompt search. *CoRR*, abs/2206.04673, 2022. doi: 10.48550/ARXIV.2206.04673. URL https://doi.org/10.48550/arXiv.2206.04673. 3

Jiacheng Zhu, Kristjan H. Greenewald, Kimia Nadjahi, Haitz Sáez de Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. 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=txRZBD8tBV. 9

A Related Work

Parameter efficient adapters. PEFT techniques address the high computational cost of fine-tuning large foundational models by updating only a small set of parameters, rather than the full original network. A prominent class of PEFT methods is adapter-based: small trainable modules are added to a frozen ed model. Early work inserted bottleneck adapters between layers to enable task-specific tuning without altering original weights (Houlsby et al., 2019; Rebuffi et al., 2017); later designs placed adapters in parallel to existing layers for improved adaptation (He et al., 2022a). Recent work have explored structured parameterizations, e.g., Kronecker-factored matrices (Mahabadi et al., 2021). Li et al. (2024) employ block-specific adapter designs, dynamic parameter sharing, and mixtures of experts to improve efficiency and generalization. At the matrix level, LoRA and its variants constrain weight updates to a low-dimensional subspace for memory and compute-efficient tuning (Hu et al., 2022; Liu et al., 2024). Indeed, He et al. (2022a) show that many PEFT methods can be viewed through a unified lens of adapter.

Beyond architectural modifications, other PEFT strategies focus on minimizing the number of updated weights directly. These include sparse update methods (Guo et al., 2021; Sung et al., 2021; He et al., 2022b), which identify and tune only the most critical parameters. Recent work has even explored extremely low-precision adapters through quantization (Jie et al., 2023), demonstrating that 1-bit adapters can rival or surpass other PEFT strategies in both parameter efficiency and performance.

LoRA methods and insights. Among PEFT techniques, LoRA-based methods have emerged as particularly prominent due to their simplicity, inspiring a wide range of follow-up studies.

Several works aim to improve LoRA's design. Some focus on alternative initialization schemes. PiSSA (Meng et al., 2024) and CorDA (Yang et al., 2024) leverage spectral decompositions of the pretrained weights to initialize LoRA modules more effectively. Shuttleworth et al. (2024) observe that LoRA introduces novel singular directions absent in full fine-tuning. Building on this, LoRA-Null (Tang et al., 2025) initializes adapters in the nullspace of pretrained activations to reduce forgetting. Other approaches propose architectural modifications. DoRA (Liu et al., 2024) decomposes pretrained weights into basis and scaling components and applies LoRA on the basis. VeRA (Kopiczko et al., 2024) further simplifies this by fixing both A and B to random bases and learning only scaling coefficients. RandLoRA (Albert et al., 2025) aggregates multiple VeRA-like components to achieve higher-rank updates. HiRA (Huang et al., 2025) follows a different route, applying element-wise multiplication between the LoRA module and the pretrained weight. These methods are all motivated by structural properties of the pretrained weights.

A parallel line of work investigates LoRA's learning behavior. Hayou et al. (2024b,a) analyze how imbalanced initialization affects feature-level dynamics during training. Zhu et al. (2024) report an asymmetry between the down- and up-projection matrices induced by standard initialization, which motivates subsequent variants such as HydraLoRA (Tian et al., 2024) and MALoRA (Wang et al., 2025). We refer the reader to Mao et al. (2025); Han et al. (2024) for more comprehensive overviews of LoRA and its many variants.

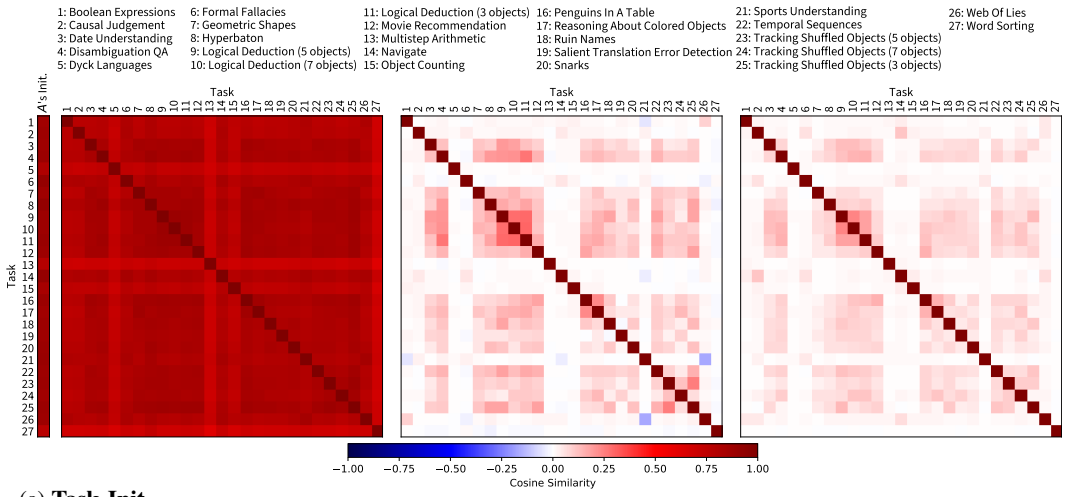
Our method differs from prior architectural improvements in that it also analyzes the input features to the target layers, rather than focusing solely on the pretrained weights. Drawing inspiration from studies on LoRA's learning behavior, our approach introduces a feature-aware projection objective that preserves information in the input representation before applying the low-rank update.

B Empirical Study on LoRA's Asymmetric Learning Behavior

To empirically illustrate this asymmetry, we conduct an adaptation experiment across multiple tasks. Following Huang et al. (2023), we choose the few-shot adaptation setting on the BIG-Bench Hard benchmark (BBH; Suzgun et al., 2023), which comprises 27 diverse tasks. We use Flan-T5 (Chung et al., 2024) as the base pretrained model. For each task j, we either fully fine-tune the pretrained model or learning LoRA adapters on a set of target layer Λ for a fixed number of steps T, reaching zero training loss in both cases. All LoRA adapters are initialized with the same random seed across tasks, ensuring that $A_{0,j}^{(\ell)} = A_0^{(\ell)}$ for every target layer $\ell \in \Lambda$. This facilitates comparison of the learned LoRA matrices across tasks.

To analyze inter-task similarity, across all target layers Λ, we flatten and concatenate full-fine-tune updates and the trained LoRA matrices, yielding vectors for each task j: $\theta_{A,j} = \big\|_{\ell\in\Lambda} \mathrm{vec}\big(A^{(\ell)}_{T,j}\big)$, $\theta_{B,j} = \big\|_{\ell\in\Lambda} \mathrm{vec}\big(B^{(\ell)}_{T,j}\big)$, and $\Delta\theta_{W,j} = \big\|_{\ell\in\Lambda} \mathrm{vec}\big(W^{(\ell)}_{T,j} - W^{(\ell)}\big)$. Fig. 3 then presents cosine-similarity matrices for two cases: in panel (a) ("Task–Init, LoRA-A") we compare each trained vector $\theta_{A,j}$ to their common LoRA-A initialization; panels (b)–(d) ("Task–Task, LoRA-A", LoRA-B and Full FT, respectively) show pairwise similarities $\cos(\theta_{A,i}, \theta_{A,j})$, $\cos(\theta_{B,i}, \theta_{B,j})$, and $\cos(\Delta\theta_{W,i}, \Delta\theta_{W,j})$.

Remarkably, Fig. 3a shows that A matrices are still pretty similar to their initialization, while Fig. 3b is largely uniform across tasks. This indicates that the learned A matrices undergo little change during adaptation and capture minimal task-dependent variation. In contrast, Figures 3c and 3d reveal nearly identical block structures, suggesting that the task-specific information recovered by full fine-tuning is almost entirely absorbed by the B matrices.



(a) **Task-Init** LoRA-A (b) **Task-Task**, LoRA-A (c) **Task-Task**, LoRA-B (d) **Task-Task**, Full FT update

Figure 3: **Cosine-similarity matrices for LoRA and full fine-tune updates on BIG-Bench Hard tasks.** (a) shows the similarity between each trained LoRA-A vector and its initialization; panels (b-d) show pairwise task–task similarities for LoRA-A, LoRA-B, and full fine-tune updates, respectively.

C Experimental Detail

C.1 Hyperparameters

The hyperparameters used across all models are summarized as follows. For instruction-following tasks, we adopt a batch size of 16, aligning with Hu et al. (2023) and Liu et al. (2024). For open-vocabulary image classification, we use a batch size of 64.

We use a learning rate of 3×10^{-4} for LLAMA-2 7B, and 1×10^{-4} for LLAMA-3 8B, QWEN-2.5 7B, and GEMMA-3 4B. For all, LoRA and DoRA use a scaling factor ($\lambda = \frac{\alpha}{d_h}$) of 2, while IPA uses 0.25, except for GEMMA-3 4B, where it is 0.4. For SIGLIP 2, we apply a learning rate of 1×10^{-3}, scaling factors of 2 (LoRA/DoRA) and 0.5 (IPA), with a dropout rate of 0.1 across all variants.

C.2 Ablation Studies

All ablations use LLAMA-3 8B on the instruction-following fine-tuning task (see Section 3.1).

Projector pretraining algorithm. As introduced in Section 2.4, we compare two online algorithms for estimating the top principal components: IPCA and GHA. Both optimize the same autoencoding objective eq. (2). Table 3 reports results with and without projector fine-tuning. Across all settings,

IPA-IPCA achieves higher downstream accuracy and converges more reliably than its GHA-based counterpart, making it our default choice. Detailed per-task results are provided in Table 3.

Table 3: Comparison of instruction-following answer accuracy (%) between IPCA and GHA algorithms on commonsense reasoning benchmark.

Method	Proj. FT	BoolQ	PIQA	SocialIQA	HellaSwag	WinoGrande	ARC-easy	ARC-challenge	OpenbookQA	Avg.
IPA-IPCA	✗	**74.8**	**88.6**	**81.1**	**95.4**	**85.6**	**91.7**	79.9	**87.8**	**85.6**
IPA-GHA	✗	73.3	88.1	80.3	95.0	85.1	91.0	**80.0**	87.2	85.0
IPA-IPCA	✓	**75.0**	**89.9**	81.2	**96.0**	85.9	**91.2**	79.6	**88.4**	**85.9**
IPA-GHA	✓	74.9	89.3	**81.3**	95.8	**86.3**	90.4	**80.1**	86.2	85.6

Projector pretraining set size. The `commonsense-170k` dataset is large enough to investigate how the size of the projector pretraining set affects downstream performance. In Fig. 4b, we pretrain the projector on randomly shuffled subsets ranging from 1% to 100% of the data, using a fixed seed for reproducibility. We select the first X% of examples from the shuffled split. Although performance generally improves up to around 10% of the data, we observe mitigated results beyond that point, which is likely due to variance in sample composition and/or randomized version of IPCA. Pretraining the feature projector on the full feature set takes roughly 1.7 hours on a NVIDIA H100 GPU, which is about ten times longer than using a 10% subset (≈10 minutes). Note that adapter tuning on the full dataset requires about 5 hours for 3 epochs. Despite the substantially lower cost, the full dataset yields negligible or no accuracy improvement (and occasionally slight degradation due to variance), so we conclude that 10% is a practical sweet spot for efficient pretraining on `commonsense-170k` dataset without sacrificing downstream performance. Detailed results are provided in Table 5.

Projected feature dimension. In our ablation study, we vary the hidden dimension d_h for IPA, LoRA, and DoRA, while keeping the learning rate, pretraining set size, and scaling ratio fixed. Fig. 4a shows a characteristic bell-shaped curve for both IPA and LoRA: accuracy falls off steeply at very low dimensions, reaches a maximum over an intermediate range, then gradually declines as d_h increases further. Importantly, IPA is more robust than LoRA: at $d_h = 8$, it matches LoRA's performance at $d_h = 16$, whereas LoRA's accuracy drops sharply. DoRA maintains a relatively flat performance profile across all tested dimensions but underperforms IPA once $d_h \geq 8$. For intermediate dimensions ($d_h = 16, 32, 64$), LoRA still outperforms DoRA. Detailed per-task results are provided in Table 4.

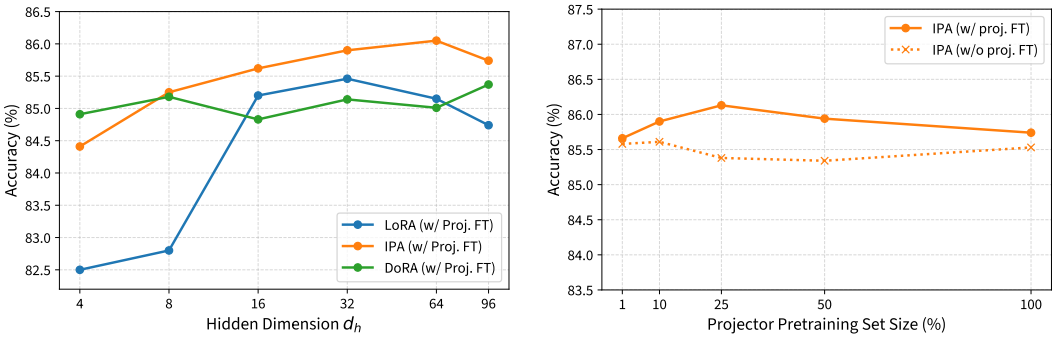


Figure 4: Average accuracy of LLAMA-3 8B models fine-tuned on commonsense benchmark with (a) varying hidden dimension d_h for IPA, compared to LoRA and DoRA, both with input projection fine-tuning •••, and (b) IPA (with projection fine-tuning • or without ×) with varying percentage of the training dataset to obtain the projection pretraining feature set.

Tables 4 and 5 show the detailed results of the ablation studies in Figs. 4a and 4b in Appendix C.2.

Table 4: Detailed results of the ablation study on different hidden dimensions.

Method	Proj. FT	Hidden Dim.	BoolQ	PIQA	SocialIQA	HellaSwag	WinoGrande	ARC-easy	ARC-challenge	OpenbookQA	Avg.
LoRA	✓	4	62.1	87.9	78.9	91.3	84.0	89.9	79.4	86.6	82.5
		8	62.1	88.8	80.5	92.3	83.0	90.2	80.7	84.8	82.8
		16	74.7	87.4	80.9	95.4	86.7	90.0	79.4	87.2	85.2
		32	75.4	88.6	80.7	95.4	86.2	91.2	80.1	86.1	85.5
		64	75.1	88.4	81.0	93.0	86.9	90.4	79.7	86.8	85.1
		96	74.9	88.4	79.8	94.6	86.3	89.6	78.8	85.4	84.7
DoRA	✓	4	73.6	88.6	79.8	95.5	85.1	90.2	80.3	86.2	84.9
		8	75.6	89.1	80.7	95.6	85.2	90.9	78.7	85.8	85.2
		16	73.5	88.9	80.2	95.3	86.1	90.5	78.6	85.6	84.8
		32	75.3	89.3	80.8	95.3	85.8	89.9	79.3	85.6	85.1
		64	74.8	88.6	80.9	94.9	85.3	89.4	79.9	86.2	85.0
		96	74.6	89.0	80.0	95.3	85.9	90.4	79.0	88.8	85.4
IPA	✓	4	73.7	88.0	79.2	95.0	84.0	89.9	79.7	85.8	84.4
		8	73.7	89.0	81.1	95.6	86.3	91.0	80.1	85.2	85.2
		16	74.6	88.9	80.6	96.0	85.1	91.0	80.3	88.6	85.6
		32	75.0	89.9	81.2	96.0	85.9	91.2	79.6	88.4	85.9
		64	75.9	88.4	80.4	95.9	87.5	91.5	81.0	87.8	86.1
		96	75.6	88.2	81.4	95.9	86.6	91.0	80.5	86.8	85.7

Table 5: Detailed results of the ablation study on projector pretraining set size.

Method	Proj. FT	Proj. Pre-training Set	BoolQ	PIQA	SocialIQA	HellaSwag	WinoGrande	ARC-easy	ARC-challenge	OpenbookQA	Avg.
IPA	✓	1%	75.2	88.8	81.0	95.6	86.5	91.3	79.6	87.2	85.7
		10%	75.0	89.9	81.2	96.0	85.9	91.2	79.6	88.4	85.9
		25%	75.4	89.4	81.8	96.0	88.1	91.1	79.9	87.4	86.1
		50%	74.9	89.2	81.5	95.9	87.6	91.1	80.7	86.6	85.9
		100%	75.1	88.8	80.8	96.1	86.9	90.9	79.9	87.6	85.7
	✗	1%	74.1	88.5	80.9	95.3	86.1	91.4	80.8	87.6	85.6
		10%	74.9	88.5	81.0	95.7	85.6	91.0	80.0	88.2	85.6
		25%	73.6	88.2	80.5	95.5	85.8	91.0	80.1	88.4	85.4
		50%	74.3	88.2	80.7	95.3	85.4	90.2	80.4	88.2	85.3
		100%	73.7	88.0	81.1	95.2	86.6	90.7	80.1	88.8	85.5