On the Importance of Trivial Baselines: Re-evaluating LoRA Adapter Transfer for Generative Tasks

Andreas Hochlehnert [1] **Danni Liu** [2] **Jan Niehues** [2] **Shankar Kumar** [1] **Felix Stahlberg** [1]

Abstract

Modern LLM post-training is becoming increasingly compute-intensive, making the re-alignment and fine-tuning for every new base model iteration computationally unsustainable. While training-free techniques for transferring post-training weights such as LoRA adapters across different base model versions exist, their effectiveness remains under-explored. In this work, we systematically evaluate the efficacy of LoRA weight transfer methods, ranging from simple arithmetic weight transfer to more complex frameworks like CrossLoRA and ProLoRA. We assess performance across a diverse suite of benchmarks, including discriminative tasks (ARC-Easy/Challenge) and multi-token generation (machine translation – WMT19). Our results demonstrate that directly copying LoRA adapters between related base models consistently outperforms more elaborate transfer schemes. However, we identify a significant disparity in transfer robustness: while Multiple Choice Question Answering (MCQA) capabilities are preserved with relative ease, generation performance suffers substantially. Furthermore, we establish that transfer success depends on the similarity between the source and the target base model weights.

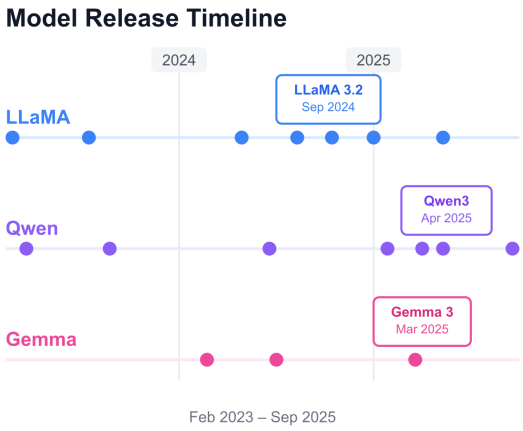


Figure 1. Release timeline of major open-weight language model families. The details are listed in Table 1.

1. Introduction

The rapid evolution of large language models (LLMs) has enabled dramatic improvements in natural language understanding and generation, but this progress has also introduced significant computational challenges. Today's LLMs contain hundreds of billions to trillions of parameters and are pretrained on vast corpora. Post-training often involves compute-intensive steps such as instruction tuning, complex reasoning alignment, and task-specific model adaptation.

Parameter-Efficient Fine-Tuning (PEFT) methods such as LoRA (Hu et al., 2022; Schulman & Lab, 2025) effectively reduce the computational overhead of task adaptation. However, a significant practical challenge in post-training is the rapid evolution of base LLMs. Major model versions are often released in 3-month cycles (Fig. 1). Traditionally, migrating to a newer base model requires the complete re-training of all associated LoRA adapters – an upgrade path that becomes computationally prohibitive with an increase in the frequency of model releases and the number of task-specific adapters. Recent research (Xia et al., 2025; Farhadzadeh et al., 2025b) has begun to explore the transferability of LoRA weights between base model iterations without re-training. However, the effectiveness of these techniques has not been systematically mapped. In particular, it remains unclear how transfer robustness varies between discriminative tasks and multi-token generation.

In this work, we systematically evaluate the transferability of LoRA adapters across different base models, using techniques that range from simple arithmetic weight transfer to projection-based methods. Our key contributions are:

[1]Google Research [2]Karlsruhe Institute of Technology. Correspondence to: Andreas Hochlehnert <andreas.hochlehnert@bethgelab.org>.

Presented at the ICML 2026 Workshop "Continual Adaptation at Scale: Towards Sustainable AI". Copyright 2026 by the author(s).

Contributions

Trivial LoRA transfer is a strong baseline. We show that directly copying LoRA adapters between related base LLMs consistently outperforms more elaborate transfer schemes on discriminative and generative tasks, including Cross-LoRA, ProLoRA, and linear and difference-based transformations. Across multiple model families and tasks, naive copy-transfer yields the most stable improvements over the unadapted base model, underscoring the importance of strong trivial baselines when evaluating adapter transfer methods.

Transfer success is governed by source–target model proximity. We demonstrate that the effectiveness of LoRA weight copying is tightly controlled by the similarity of the underlying base models. Using the distance between the model parameters, we uncover a strong correlation between the model similarity and the transfer performance, explaining the variability observed across different model families.

LoRA transfer favors tasks requiring minimal model adaptation. We find that LoRA weight copying is robust for tasks that induce only minor parameter updates, such as Multiple Choice QA. In contrast, the performance degrades substantially for tasks that require more extensive model adaptation such as machine translation. This task-dependent divergence explains why adapter transfer succeeds for discriminative benchmarks but fails for complex generative settings.

2. Methodology

This section details the methods we use to transfer LoRA adapters. Let $\mathbf{W}_\text{s}, \mathbf{W}_\text{t} \in \mathbb{R}^{d_\text{in} \times d_\text{out}}$ be the corresponding weight matrices of the source and the target models, respectively, for a specific module (e.g., the query-projection in the attention). LoRA training on the source model yields the source adapter $\Delta\mathbf{W}_\text{s} = \mathbf{B}_\text{s}\mathbf{A}_\text{s}$, where $\mathbf{B}_\text{s} \in \mathbb{R}^{d_\text{in} \times r}$ and $\mathbf{A}_\text{s} \in \mathbb{R}^{r \times d_\text{out}}$. Our goal is to find a target LoRA adapter $\Delta\mathbf{W}_\text{t} = \mathbf{B}_\text{t}\mathbf{A}_\text{t}$ (with rank r) that works with $\mathbf{W}_\text{t}$.

In addition to the intricate methods from the literature Cross-LoRA (Xia et al., 2025) and ProLoRA (Farhadzadeh et al., 2025b) (details on these methods in Appendix C.1), we explore the following simple arithmetic methods:

SimpleCopy Our most trivial method is the direct unaltered copy of the LoRA weights from the source model to the target model:

$$\mathbf{B}_\text{t}\mathbf{A}_\text{t} = \Delta\mathbf{W}_\text{t} = \Delta\mathbf{W}_\text{s} = \mathbf{B}_\text{s}\mathbf{A}_\text{s}$$

A similar approach was explored by Farhadzadeh et al. (2025b) for diffusion models.

SimpleDifference This method builds on weight-space arithmetic. It assumes that the adapter $\Delta\mathbf{W}_\text{s}$ is optimized relative to $\mathbf{W}_\text{s}$, and to apply it to $\mathbf{W}_\text{t}$, we must "re-center" it by accounting for the "base model drift" ($\mathbf{W}_\text{t} - \mathbf{W}_\text{s}$).

$$\mathbf{B}_\text{t}\mathbf{A}_\text{t} = (\mathbf{W}_\text{t} - \mathbf{W}_\text{s}) + \mathbf{B}_\text{s}\mathbf{A}_\text{s}$$

SimpleLinear This method, common in embedding space alignment (Mikolov et al., 2013), assumes the source and target weight spaces are linearly related. We first learn an optimal linear transformation matrix $\mathbf{P} \in \mathbb{R}^{d_\text{in} \times d_\text{in}}$ that maps the source weights $\mathbf{W}_s$ to the target weights $\mathbf{W}_t$ by solving a linear least-squares problem:

$$\mathbf{P} = \arg\min_{\mathbf{P}} ||\mathbf{P}\mathbf{W}_\text{s} - \mathbf{W}_\text{t}||_\text{F}^2$$

The solution is given by $\mathbf{P} = \mathbf{W}_\text{t}\mathbf{W}_\text{s}^+$, where $\mathbf{W}_\text{s}^+$ is the Moore-Penrose pseudoinverse (Penrose, 1955) of $\mathbf{W}_\text{s}$. The core hypothesis is that this $\mathbf{P}$, which aligns the base models, can also align the task adapters.

3. Experiments

We test transferability across *discriminative* and *generative* tasks, specifically MCQA and multi-token generation. This allows us to compare the performances when the model is adapted to contrasting functional requirements. The training datasets are used to learn the source weights ($\Delta\mathbf{W}_s$), while evaluation is performed on the target weights ($\mathbf{W}_t + \Delta\mathbf{W}_t$). Further details are provided in Appendix D.2.

Discriminative Tasks (MCQA) We utilize **ARC-C/E** (Clark et al., 2018) for both adapter training and evaluation, following standard MCQA fine-tuning protocols. These tasks represent common sense reasoning and knowledge retrieval. These tasks require only single-token generation, minimizing error-accumulation.

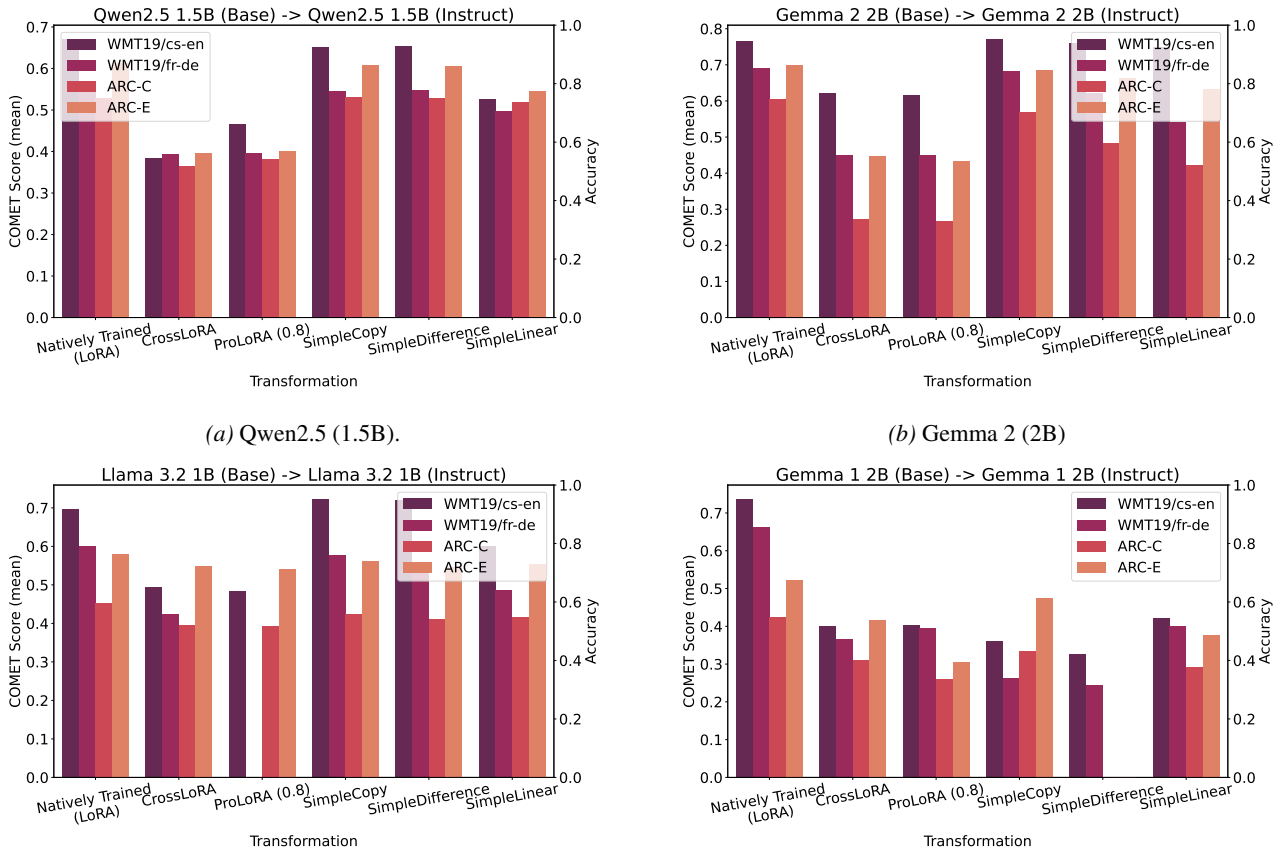
Generative Tasks (Translation) We focus on machine translation using the **WMT19** benchmark (Barrault et al., 2019). Specifically, we train and evaluate on the Czech-English (cs-en) and French-German (fr-de) subsets to test cross-lingual transfer capabilities.

We investigate transferability from the *Base* to *Instruct* variants of four open-weights models: Llama 3.2 (2B), Qwen 2.5 (1.5B), Gemma (2B), and Gemma 2 (2B). See Appendix D.1 for details.

4. Results

4.1. Trivial Transfer Outperforms Complex Methods

Fig. 2 compares the performance of *SimpleCopy*, *SimpleDifference*, and *SimpleLinear* with the more intricate Cross-LoRA and ProLoRA frameworks. The performance of the four unadapted base models are shown in Appendix E.1. The latter two methods exhibit significant volatility. While they match the performance of natively trained LoRA adapters for Llama 3.2 on ARC tasks, they suffer substantial degradation across other model families and benchmarks. Notably, these complex methods frequently fail for translation tasks, especially into non-English languages.



(a) Qwen2.5 (1.5B).

(b) Gemma 2 (2B)

(c) Llama 3.2 (1B).

(d) Gemma 1 (2B)

Figure 2. Comparison between different transfer methods. The trivial *SimpleCopy* method consistently outperforms more complex frameworks like CrossLoRA and ProLoRA across the board. These sophisticated approaches struggle significantly with translation tasks – especially when targeting non-English languages – whereas simpler methods tend to be more stable.

In contrast, the simpler transfer methods frequently match natively trained LoRA adapters on ARC-C and ARC-E. This suggests that basic weight transfer between pre-trained and instruction-tuned models is indeed viable for MCQA tasks. While these methods also experience performance losses on the translation tasks, the degradation is less severe than that observed with ProLoRA and CrossLoRA. Most importantly, our most basic method, *SimpleCopy*, emerged as the most stable method throughout our study. This finding is particularly noteworthy in light of the results reported by Farhadzadeh et al. (2025b), who observed that direct weight copying does not always match ProLoRA on diffusion models. The efficacy of transfer methods appears to be highly contingent on model architecture and does not generalize across model types. This highlights a broader necessity in the field: the inclusion of trivial baselines and a diverse set of models is essential to contextualize the performance of novel, more complex methods.

We will focus our subsequent analyses exclusively on *SimpleCopy* as it proved the most stable across our evaluations.

4.2. Source-target Model Similarity Determines the Efficacy of LoRA Weight Transfer

Existing research on LoRA adapter transfer has largely overlooked the degree of divergence between source and target models (Xia et al., 2025; Farhadzadeh et al., 2025b). However, the distance between the pre-trained model and its instruction-tuned counterpart varies significantly across different model families. To quantify the degree of divergence between a source model and a target model, we measure their distance directly in the weight space. For each pair of parameter source tensor $W_{\mathrm{s}}^{(i)}$ and the corresponding target tensor $W_{\mathrm{t}}^{(i)}$, we compute the ℓ_2-distance between them:

$$d^{(i)} = \frac{\|W_s^{(i)} - W_t^{(i)}\|_2}{|W_t^{(i)}|},$$

where $|W_t^{(i)}|$ denotes the number of elements in $W_t^{(i)}$. The overall source-target model distance is then defined as the mean $d^{(i)}$ across all parameters. This yields a single scalar that captures the deviation between the two models, while

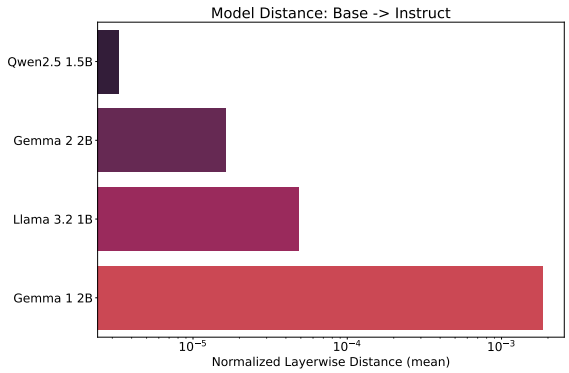


Figure 3. Distance between the pre-trained and instruction-tuned models.

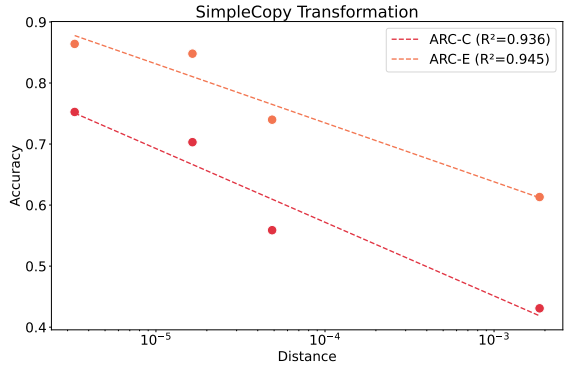


Figure 4. Inverse relationship between the efficacy of *SimpleCopy* and the source-target model distance on ARC-E and ARC-C. The dashed lines indicate a least-squares linear fit in log-distance space.

preserving layerwise statistics for further analysis.

As illustrated in Fig. 3, the pre-trained and instruction-tuned Qwen 2.5 (1.5B) models are closely aligned, but the corresponding difference for Gemma 1 is about 560 times higher. Fig. 2d shows that transfer methods suffer most degradation on Gemma 1, particularly on generation tasks.

Intuitively, successful LoRA weight transfer requires high source-target model alignment. Fig. 4 shows a strong inverse correlation between the source-target model distance with the accuracy on ARC-C and ARC-E for the *SimpleCopy* method: performance degrades sharply as model distance increases, suggesting that direct weight transfer lacks robustness when models are significantly misaligned. For the correlation coefficients and a similar analysis for the translation tasks see Appendix E.2

4.3. LoRA Weight Transfer is Most Effective for Tasks Requiring Minimal Model Adaptation

We find that transferability is also governed by the extent to which the base model needs to be adapted to a specific task.

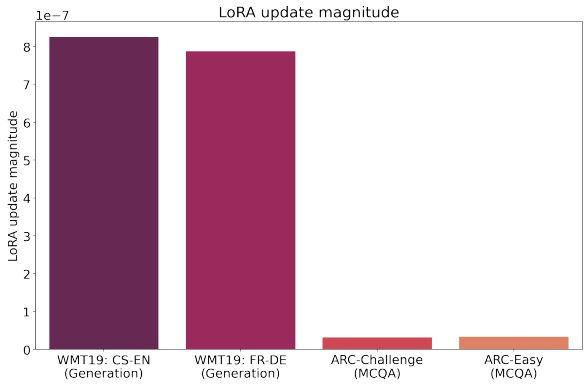


Figure 5. Training on ARC-C and ARC-E yields only marginal adjustments to the base model, whereas translation tasks necessitate substantially larger LoRA adaptation.

We quantify this factor using the LoRA update magnitude, defined as the norm of the product of the adapter matrices (we use the notation from Sec. 2):

$$||\Delta \mathbf{W}||_{\mathrm{F}}^2 = ||\mathbf{BA}||_{\mathrm{F}}^2$$

As illustrated in Fig. 5, translation tasks demand more extensive adaptation of the base model compared to MCQA classification. Translation requires the autoregressive generation of long token sequences, whereas classification is restricted to a single-token output. Consequently, classification tasks are less susceptible to error accumulation over multiple decoding steps. This difference explains why LoRA weight transfer depends heavily on the task: Fig. 2 shows that while MCQA tasks handle the transferred weights relatively well, translation tasks suffer a much worse drop in performance. Appendix E.3 further elaborates on this effect.

5. Conclusion

In this work, we revisit the problem of training-free LoRA adapter transfer across evolving base language models. Through a systematic evaluation spanning multiple model families, transfer methods, and task types (discriminative and generative), we showed that a trivial baseline – directly copying LoRA adapters, consistently outperforms or matches more elaborate transfer schemes. Our results highlight two key factors governing transfer success: the geometric proximity between source and target base models, and the degree of task-specific adaptation required. While LoRA transfer proves robust for lightweight discriminative tasks such as MCQA, it degrades substantially for multi-token generative settings like machine translation. Together, these findings emphasize the importance of strong trivial baselines and caution against overgeneralizing positive transfer results across tasks. We hope this study highlights the need for a stronger empirical foundation and more robust evaluation methods in future adapter transfer research.

Acknowledgments

Part of this work was funded by the KiKIT (The Pilot Program for Core-Informatics at the KIT) of the Helmholtz Association. Part of this work was performed on the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

References

Barrault, L., Bojar, O., Costa-jussà, M. R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., Müller, M., Pal, S., Post, M., and Zampieri, M. Findings of the 2019 conference on machine translation (WMT19). In Bojar, O., Chatterjee, R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Martins, A., Monz, C., Negri, M., Névéol, A., Neves, M., Post, M., Turchi, M., and Verspoor, K. (eds.), *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 1–61, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5301.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

Farhadzadeh, F., Das, D., Borse, S., and Porikli, F. LoRA-X: Bridging foundation models with training-free cross-model adaptation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a.

Farhadzadeh, F., Das, D., Borse, S., and Porikli, F. Zero-shot adaptation of parameter-efficient fine-tuning in diffusion models. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025b.

Golub, G. H. and Van Loan, C. F. *Matrix computations*. JHU press, 2013.

Guha, E., Marten, R., Keh, S., Raoof, N., Smyrnis, G., Bansal, H., Nezhurina, M., Mercat, J., Vu, T., Sprague, Z., Suvarna, A., Feuer, B., Chen, L., Khan, Z., Frankel, E., Grover, S., Choi, C., Muennighoff, N., Su, S., Zhao, W., Yang, J., Pimpalgaonkar, S., Sharma, K., Ji, C. C.-J., Deng, Y., Pratt, S., Ramanujan, V., Saad-Falcon, J., Li, J., Dave, A., Albalak, A., Arora, K., Wulfe, B., Hegde, C., Durrett, G., Oh, S., Bansal, M., Gabriel, S., Grover, A., Chang, K.-W., Shankar, V., Gokaslan, A., Merrill, M. A., Hashimoto, T., Choi, Y., Jitsev, J., Heckel, R., Sathiamoorthy, M., Dimakis, A. G., and Schmidt, L. Openthoughts: Data recipes for reasoning models, 2025. URL https://arxiv.org/abs/2506.04178.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Jindal, I., Badrinath, C., Bharti, P., Vinay, L., and Sharma, S. D. Balancing continuous pre-training and instruction fine-tuning: Optimizing instruction-following in LLMs. *CoRR*, abs/2410.10739, 2024. doi: 10.48550/ARXIV.2410.10739.

Li, M., Nie, Z., Zhang, Y., Long, D., Zhang, R., and Xie, P. Improving general text embedding model: Tackling task conflict and data imbalance through model merging. *CoRR*, abs/2410.15035, 2024. doi: 10.48550/ARXIV.2410.15035.

Li, Y., Meng, F., Zhang, M., Zhu, S., Wang, S., and Xu, M. LoRASuite: Efficient LoRA adaptation across large language model upgrades. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Lin, P.-J., Balasubramanian, R., Liu, F., Kandpal, N., and Vu, T. Efficient model development through fine-tuning transfer. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2617–2636, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.131.

Mikolov, T., Le, Q. V., and Sutskever, I. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

Penrose, R. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955. doi: 10.1017/S0305004100030401.

Ran, L., Cun, X., Liu, J.-W., Zhao, R., Zijie, S., Wang, X., Keppo, J., and Shou, M. Z. X-Adapter: Adding universal compatibility of plugins for upgraded diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8775–8784, June 2024.

Rei, R., Stewart, C., Farinha, A. C., and Lavie, A. COMET: A neural framework for MT evaluation. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2685–2702, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.213. URL https://aclanthology.org/2020.emnlp-main.213/.

Schulman, J. and Lab, T. M. LoRA without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

Wang, R., Ghosh, S., Cox, D. D., Antognini, D., Oliva, A., Feris, R., and Karlinsky, L. Trans-LoRA: Towards data-free transferable parameter efficient finetuning. In *NeurIPS*, 2024.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Lopes, R. G., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., and Schmidt, L. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 2022.

Xia, F., Liao, M., Fang, Y., Li, D., Xie, Y., Li, W., Li, Y., Xia, D., and Huang, J. Cross-LoRA: A data-free LoRA transfer framework across heterogeneous LLMs. *arXiv preprint arXiv:2508.05232*, 2025.

Yadav, P., Tam, D., Choshen, L., Raffel, C. A., and Bansal, M. TIES-Merging: Resolving interference when merging models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (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*, 2023.

A. Related Work

Transfer Approaches Although recent work has introduced various methods for LoRA transfer, they have concentrated on discriminative NLP tasks (Wang et al., 2024; Xia et al., 2025; Li et al., 2025) or image generation (Ran et al., 2024; Farhadzadeh et al., 2025a;b). While Farhadzadeh et al. (2025b) reported initial results on text generation, their study was restricted to transfer between checkpoints close in the same training run. Therefore, LoRA transferability across different models for autoregressive text generation remains underexplored. The sequential nature of text generation, which requires mapping representations to discrete tokens at every step, may pose additional challenges for transfer. Existing transfer approaches vary in their implementation overhead and portability. Several require auxiliary steps, such as distillation from synthetic data (Wang et al., 2024), training additional models to predict feature mappings (Ran et al., 2024), or fine-tuning on the target task (Li et al., 2025). These prerequisites reintroduce training costs, potentially diminishing the efficiency benefits of PEFT. Other methods depend on specialized adapter architectures (Farhadzadeh et al., 2025a), which limits their generality given the ubiquity of LoRA. In light of these considerations, we focus on training-free approaches that are compatible with standard LoRA, namely Cross-LoRA (Xia et al., 2025) and ProLoRA (Farhadzadeh et al., 2025b).

Degree of Adaptation and Task Vectors Recent studies suggest that the directions of weight update after fine-tuning, or task vectors (Ilharco et al., 2023) encapsulate the specific capabilities required for a downstream task. These vectors exhibit compositional properties (Wortsman et al., 2022; Ilharco et al., 2023; Yadav et al., 2023). For instance, task vectors representing the difference between instruction-tuned and base models can be algebraically combined to inject instruction-following abilities in other models (Jindal et al., 2024; Lin et al., 2025). Conceptually, LoRA operates on the same principle. Therefore, our study on transferring LoRA adapters is a special case of transferring task vectors across models. The geometry of these vectors is linked to training dynamics. Li et al. (2024) observed a linear relationship between the norm of the task vector and the volume of training data, suggesting that resulting adapters encode the intensity of the adaptation.

B. Model Release Timelines

Table 1 shows the chronological evolution of LLaMA, Qwen, and Gemma models. The individual markers in Fig. 1 correspond to the publicly released model generations listed in this table.

Model Family	Version	Release Date
LLaMA	LLaMA 1	Feb 2023
	LLaMA 2	Jul 2023
	LLaMA 3	Apr 2024
	LLaMA 3.1	Jul 2024
	LLaMA 3.2	Sep 2024
	LLaMA 4	Apr 2025
Qwen	Qwen (Beta)	Apr 2023
	Qwen (Public)	Sep 2023
	Qwen 2	Jun 2024
	Qwen 2.5	Jan 2025
	Qwen 3	Apr 2025
Gemma	Gemma 1	Feb 2024
	Gemma 2	Jun 2024
	Gemma 3	Mar 2025

Table 1. Release timeline of major open-weight language model families.

C. Extended Methodology

C.1. Intricate Baselines

We compare our methods based on simple arithmetics with the following two more intricate baselines from the literature:

Cross-LoRA (Xia et al., 2025) This method aligns the source adapter to the target model's principal weight-space components. It involves two steps for each adapted weight matrix:

- **Decomposition:** The target weight matrix $\mathbf{W}_t$ is decomposed using Singular Value Decomposition (SVD) (Golub & Van Loan, 2013) to find its left and right singular vectors: $\mathbf{W}_t \approx \mathbf{U}_t \mathbf{\Sigma}_t \mathbf{V}_t^\top$.
- **Projection:** The source LoRA matrices $\mathbf{B}_s$ and $\mathbf{A}_s$ are independently projected into the target's singular spaces. The new $\mathbf{B}_t$ is formed by projecting $\mathbf{B}_s$ into the left singular space ($\mathbf{U}_t$), and the new $\mathbf{A}_t$ is formed by projecting $\mathbf{A}_s$ into the right singular space ($\mathbf{V}_t$). This yields the transformed adapters:

$$\mathbf{B}_t = \mathbf{U}_t(\mathbf{U}_t^\top \mathbf{B}_s), \quad \mathbf{A}_t = (\mathbf{A}_s \mathbf{V}_t)\mathbf{V}_t^\top$$

ProLoRA (Farhadzadeh et al., 2025b) Similar to Cross-LoRA, this method follows a decomposition-projection pipeline, but incorporates an additional weight decomposition into orthogonal components. The source adapter $\Delta\mathbf{W}_s$ is first decomposed into its projections onto the column and row spaces and their orthogonal complements of the source weights $\mathbf{W}_s$:

$$\Delta\mathbf{W}_s = \Delta\mathbf{W}_{s,\parallel} + \Delta\mathbf{W}_{s,\perp}$$

These components are then independently projected onto the corresponding spaces of the target model, where the target singular vectors $\mathbf{U}_t$ and $\mathbf{V}_t$ are partitioned according to the rank of $\mathbf{W}_t$: $\mathbf{U}_t = [\mathbf{U}_{t,\parallel}, \mathbf{U}_{t,\perp}]$ and $\mathbf{V}_t = [\mathbf{V}_{t,\parallel}, \mathbf{V}_{t,\perp}]$. The transferred adapter $\Delta\mathbf{W}_t$ is:

$$\mathbf{U}_{t,\parallel}\mathbf{U}_{t,\parallel}^\top \Delta\mathbf{W}_{s,\parallel}\mathbf{V}_{t,\parallel}^\top \mathbf{V}_{t,\parallel} + \\ \mathbf{U}_{t,\perp}\mathbf{U}_{t,\perp}^\top \Delta\mathbf{W}_{s,\perp}\mathbf{V}_{t,\perp}^\top \mathbf{V}_{t,\perp}$$

D. Experimental Details

D.1. Models and Transfer Directions

We investigate transferability between *Base* and *Instruct* variants of state-of-the-art open-weights models. This setup allows us to test transfer robustness across proximal models (where weights are geometrically aligned) and those with significant post-training divergence.

We cover four model families, testing the transfer from Base → Instruct for each (see Table 2). This setting isolates representational drift induced by post-training while controlling for architectural and capacity differences.

Source Model	Target Model
Llama 3.2 2B	Llama 3.2 2B Instruct
Qwen 2.5 1.5B	Qwen 2.5 1.5B Instruct
Gemma 2B	Gemma 2B Instruct
Gemma 2 2B	Gemma 2 2B Instruct

Table 2. Transfer pairs evaluated in this work. We test the application of base-trained adapters to instruction-tuned models.

D.2. LoRA Training Setup

Unless stated otherwise, all experiments use LoRA adapters with rank $r = 128$, applied to the MLP projection layers ($\mathbf{W}_{\text{up}}$, $\mathbf{W}_{\text{down}}$, and $\mathbf{W}_{\text{gate}}$), which has been shown to yield strong performance in prior work (Schulman & Lab, 2025). We train all adapters with a batch size of 16 and continue training beyond empirical convergence to ensure stable optimization. Specifically, we train for 10,000 steps on ARC-C/E and 100,000 steps on WMT19.

D.2.1. Training Objectives

We train all LoRA adapters using a causal language modeling objective, with task-specific input formatting and loss masking to ensure that gradients are applied only to task-relevant target tokens.

Machine Translation (WMT19). For translation tasks, each training example is formatted as

`<bos>{source_text}\n{target_text}<eos>`

The model is trained to autoregressively generate the target sequence conditioned on the source text. The training loss is computed exclusively over the target tokens, while the source tokens are masked out. This setup corresponds to standard conditional language modeling for sequence-to-sequence tasks using a decoder-only architecture.

Multiple-Choice Question Answering (ARC). For MCQA tasks, inputs consist of an instruction prompt followed by a question and four answer options:

```
<bos>You are given a question followed
by four possible answers.Choose the
correct answer by selecting the
corresponding letter.  Respond only with
the letter (A B C D) of the correct
answer.\n\nQuestion:  {question}
\nA. {A}\nB. {B}\nC. {C}\nD. {D}
\n\nAnswer:  {answer_letter}<eos>
```

where $a \in \{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}\}$ denotes the correct answer. The model is trained to predict the single answer token, and the loss is computed only on this final answer letter. All preceding tokens are masked from the loss.

This formulation restricts supervision to the minimal output required by the task, encouraging lightweight adaptation for MCQA while avoiding unnecessary changes to the base model representations.

E. Additional Results

E.1. Baselines and Transfer Methods

Model	ARC-C	ARC-E
Llama 3.2 1B Inst.	24.8	35.1
Qwen 2.5 1.5B Inst.	71.8	85.4
Gemma 1 2B Inst.	42.6	58.3
Gemma 2 2B Inst.	71.2	86.3

Table 3. MCQA accuracy of the unadapted models on ARC benchmarks.

Model	cs–en		fr–de	
	BLEU	COMET	BLEU	COMET
Llama 3.2 1B Inst.	16.3	74.2	8.3	63.5
Qwen 2.5 1.5B Inst.	15.5	78.1	11.3	72.8
Gemma 1 2B Inst.	8.7	61.7	5.2	53.9
Gemma 2 2B Inst.	20.2	82.2	7.4	70.6

Table 4. Translation performance (BLEU and COMET) of the unadapted models on WMT19.

For each model pair and task, we evaluate the LoRA transfer methods introduced in Section 2 and compare them against a natively trained LoRA adapter on the target model. All transfer methods are training-free and operate directly in the weight space without utilizing any data.

Table 3 and 4 show the performance of the four unadapted base models.

E.2. Source-target Model Similarity for Translation Tasks

We discussed in Sec. 4.2 how the source-target model similarity determines the efficacy of LoRA weight transfer on ARC-C and ARC-E.

Fig. 6 reveals a similar trend for translation tasks. However, the presence of an outlier (Qwen 2.5) in both translation directions on the left side of the plot suggests that model distance, while critical, is not the sole determinant of transferability. The correlation coefficients for Fig. 4 and Fig. 6 are listed in Table 5. Our findings underscore the necessity of evaluating a diverse array of model families to develop a comprehensive understanding of the capabilities and limitations inherent in weight transfer methodologies.

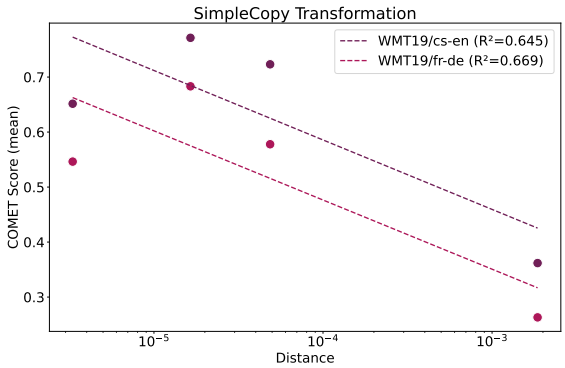


Figure 6. Inverse relationship between the efficacy of *SimpleCopy* and the source-target model distance on the translation tasks cs-en and fr-de. The dashed line indicates a least-squares linear fit in log-distance space.

Task	r	p-value
ARC-C	-0.968	3.2×10^{-2}
ARC-E	-0.972	2.8×10^{-2}
WMT19: cs–en	-0.739	1.9×10^{-1}
WMT19: fr–de	-0.651	1.8×10^{-1}

Table 5. Pearson correlation between the log source-target model distance and the performance of LoRA adapters transferred with *SimpleCopy*.

E.3. Task-dependence of LoRA Weight Transfer

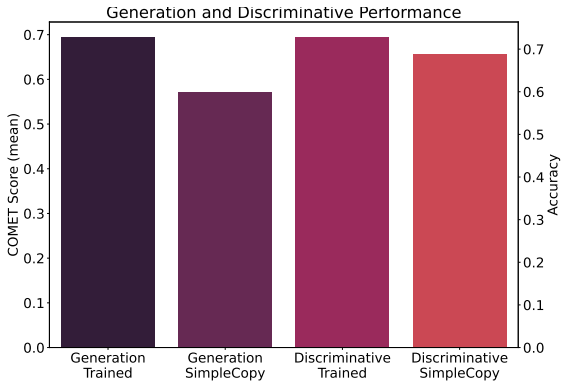


Figure 7. Average generation and discriminative performance for trained and transformed LoRA adapters using *SimpleCopy*. Discriminative tasks transfer more robustly than generative tasks. *SimpleCopy* LoRA adapters retain most of the performance of natively trained adapters on discriminative tasks, whereas generative performance degrades substantially, supporting the claim that discriminative capabilities require less task-specific adaptation.

Figure 7 highlights the task-dependent divergence in LoRA transferability. For discriminative MCQA benchmarks, *SimpleCopy* adapters preserve performance almost entirely, with average accuracy remaining effectively unchanged relative to natively trained LoRA adapters. In contrast, generative machine translation exhibits substantial degradation under the same transfer setting: the COMET scores on WMT19 drop by nearly 15% on average.

E.4. Evaluation Protocol

We evaluate all transferred and natively trained LoRA adapters in a zero-shot setting on the target models, without any additional fine-tuning. Evaluation is performed using task-appropriate metrics and consistent decoding configurations across all methods.

Machine Translation (WMT19). For translation tasks, models generate the target sentence autoregressively conditioned on the source text. Performance is measured using COMET-22 (Rei et al., 2020) on the WMT19 test sets for Czech–English and French–German.

Multiple-Choice Question Answering (ARC). For MCQA tasks, models generate a single answer token corresponding to one of the choices $\{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}\}$. We apply a simple post-processing step to extract the predicted answer letter from the model output. Performance is reported as classification accuracy over the ARC-E (Easy) and ARC-C (Challenge) evaluation sets.

E.5. Reasoning Tasks

Additionally, we investigate transferability on reasoning tasks. To this end, we train a LoRA adapter on Qwen2.5-7B (base) using OpenThoughts3-1.2M (Guha et al., 2025), and evaluate the transfer of the resulting LoRA weights to Qwen2.5-7B-Instruct, as shown in Table 6. While CrossLoRA yields only marginal improvements over the vanilla Qwen2.5-7B-Instruct baseline, SimpleCopy achieves accuracy comparable to that of the LoRA-trained Qwen2.5-7B (base) model.

Table 6. **Transfer from Qwen2.5-7B (Base) to Qwen2.5-7B-Instruct on the OpenThinker reasoning dataset.** SimpleCopy substantially outperforms CrossLoRA and achieves performance comparable to the LoRA-trained Qwen2.5-7B (Base) model.

Model	AIME24	AIME25	AMC23	GPQA Diamond	MATH500	Minerva	OlympiadBench
LoRA Trained	28.33%	22.33%	61.75%	38.23%	83.70%	36.65%	50.96%
Qwen2.5-7B-Instruct (CrossLoRA)	12.67%	4.33%	52.50%	36.36%	76.27%	36.27%	39.21%
Qwen2.5-7B-Instruct (SimpleCopy)	25.67%	20.33%	61.00%	40.96%	83.90%	37.24%	50.56%
Qwen2.5-7B-Instruct (vanilla)	12.30%	7.30%	52.80%	–	77.30%	34.90%	38.90%