LoRA Merging with SVD: Understanding Interference and Preserving Performance

Anonymous Authors[1]

Abstract

Merging Low-Rank Adaptation (LoRA) modules is a problem gaining significance as LoRA adapters proliferate. Despite various approaches showing benchmark improvements, the field lacks clear guiding principles for effective LoRA merging. Two predominant strategies exist: direct merging (DM), which preserves a memory efficient two-matrix structure but sacrifices performance, and multiplied merging (MM), which delivers superior results but abandons the memory-efficient, low-rank architecture. In this paper, we first show that DM introduces interfering cross-terms that degrade performance, while MM exhibits linear mode connectivity in the loss landscape, making it an optimal strategy for merging. Then we demonstrate that merging with an SVD-based strategy combines MM's performance advantages with DM's memory efficiency, delivering the best of both approaches.

1. Introduction

The rise of Large Language Models (LLMs) (Touvron et al., 2023; Reid et al., 2024; Achiam et al., 2023) has popularized their use as assistants for a variety of knowledge-intensive tasks. However, for some tasks, users may find that an out-of-the-box LLM is insufficient and requires additional training. Given that even the smallest usable models have billions of parameters, the computational cost of training them can be prohibitive. Thankfully, the recent rise of Parameter-Efficient Fine-Tuning (PEFT) methods – like LoRA (Hu et al., 2021) and DoRA (Liu et al., 2024) – enables LLMs to train at a fraction of the cost.

In practical applications requiring models to handle diverse queries, the development of specialized expert models for every task is often infeasible. Furthermore, employing PEFT for each task results in a number of models that scales linearly with the quantity of target tasks. Consequently, repositories like the Hugging Face Hub (Wolf et al., 2019) now host an expanding collection of these specialized PEFT modules. Serving this multitude of expert models presents significant challenges, particularly under limited GPU memory constraints. Model merging (Tang et al., 2024) aims to mitigate this limitation by consolidating multiple fine-tuned PEFT modules into a single model, that generalizes across many tasks.

Given a pre-trained base model parametrized by $\mathbf{W}$, LoRA fine-tunes the model by injecting two matrices: $\mathbf{W}+\Delta W = \mathbf{W} + \mathbf{BA}$ where $\mathbf{W}$, $\mathbf{BA} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times n}$. For LoRA merging, $\mathbf{W}$ remains consistent across all models and the adapters $\mathbf{BA}$ are trained on different tasks and then merged. In our setting, LoRA adapters are trained in image classification tasks but merging LoRA adapters can also extend to natural language processing and multimodal domains (Tang et al., 2024).

LoRA merging has generally been done in two ways. "**Direct-Merge**" (L_{DM}) directly combines $\mathbf{A}$s and $\mathbf{B}$s from different adapters separately, while "**Mutliplied-Merge**" (L_{MM}) first multiples $\mathbf{A}$ and $\mathbf{B}$ from the same adapter into $\mathbf{BA}$ before merging. For L_{MM}, the LoRA adapters lose their shape as well as any memory-efficiency from LoRA's low-rank structure (such as memory-efficient storage). Examples of L_{DM} include (Huang et al., 2023; Zhao et al., 2024; Prabhakar et al., 2024) and examples of L_{MM} include (Stoica et al., 2024; Shah et al., 2023; Wu et al., 2024).

Often, it is ambiguous which method is preferable. L_{DM} retains the low-rank matrix structure of LoRA, which enables better memory efficiency during the merging step and during storage. It also makes multi-LoRA serving cheaper as low rank matrices can be loaded and offloaded from the GPU (Yadav et al., 2023a). Additionally, in settings such as QLoRA (Dettmers et al., 2023), the shape of the original LoRA *must* be preserved, as it is challenging to merge the LoRA modules back to quantized base model weights. In contrast, L_{MM} does not preserve this low-rank structure of the LoRA matrices but often enables better performance. But *why* is there a performance gap? and is there a better way to retain LoRA's shape without performance degradation?

We strive to answer both these questions by analyzing the differences between L_{MM} and L_{DM}. Our analysis reveals that L_{DM} introduces interference terms absent in L_{MM} which severely degrade performance - when merging 8 Lo-

RAs, we observe L_{MM} outperforms L_{DM} by an $+50.15\%$ accuracy. Furthermore, we demonstrate how using SVD on top of L_{MM} can retain the memory-efficient LoRA shape with virtually no accuracy loss. We supplement this finding mathematically by demonstrating that the error from SVD will be less than the interference error in L_{DM}. Empirically, our method outperforms state-of-the-art L_{DM} approaches like LoRA LEGO by $+7.12\%$ on average. Finally, we demonstrate that L_{MM} exhibits linear mode connectivity (Frankle et al., 2019) in the loss landscape while L_{DM} does not, providing additional theoretical justification for preferring multiplied merging when combining LoRA.

2. Related Works

There are two main approaches to model-merging - data-dependent and data-free. Data-dependant approaches (Matena & Raffel, 2021; Yang et al., 2023; Prabhakar et al., 2024) use data to adjust or train the mixture of models. In our setting, we focus on the data-free setting. Model Soups (Wortsman et al., 2022) simply averages the model weights together. Task-Arithmetic merging (Ilharco et al., 2022) sums the base model with scaled task vectors (the difference between the base model and the fine-tuned model). TIES-Merging (Yadav et al., 2023b) merges models by minimizing the interference of parameters. DARE (Yu et al., 2023) uses a dropout and rescale operation. More recent model-merging methods (Matena & Raffel, 2021; Tam et al., 2023; Mavromatis et al., 2024; Lu et al., 2024; Yang et al., 2023; Feng et al., 2024; Daheim et al., 2023) have also explored fancier approaches to combining models. Methods specific to merging LoRAs include KnOTS (Stoica et al., 2024) which aligns the LoRA into a common subspace with SVD prior to merging, LoRA Soups (Prabhakar et al., 2024) which learns a linear combination of concatenated LoRAs, LoRA LEGO (Zhao et al., 2024) which clusters LoRAs before merging, and ZipLora (Shah et al., 2023) which learns optimal scaling coefficients. While these approaches are each optimal in their own specific setting (e.g. data-free, adaptable rank, post-hoc training etc), we articulate a framework for approaching LoRA-merging in general.

3. Method

3.1. Preliminaries

LoRA Fine-Tuning. The shape and structure of LoRA was described in the introduction. Without LoRA, the activation for layer $\mathbf{W}_i$ is $z_i = \mathbf{W}_i z_{i-1}$. With LoRA, this activation becomes $z_i = \mathbf{W}_i z_{i-1} + \frac{\alpha}{r}\mathbf{BA}$, where α is a scalar and r is the rank of the LoRA. During training, only $\mathbf{BA}$ is tuned and all weights $\mathbf{W}$ are frozen.

Linear Mode Connectivity. Linear Mode Connectivity (LMC) (Frankle et al., 2019) describes when two models can be effectively combined through weight interpolation. This is measured by the barrier function:

$$B(\theta_1, \theta_2) = \sup_{\alpha}[L(\alpha\theta_1 + (1-\alpha)\theta_2)] - [\alpha L(\theta_1) + (1-\alpha)L(\theta_2)]. \quad (1)$$

Here, sup indicates the supremum, θ_1 and θ_2 represent model parameters, L is the loss function, and $B(\theta_1, \theta_2)$ quantifies the maximum elevation in loss along the linear interpolation path relative to the convex combination of endpoint losses. When $B(\theta_1, \theta_2) \approx 0$, the models are LMC, indicating they share the same loss landscape basin and are ideal candidates for parameter averaging techniques (Frankle et al., 2019; Entezari et al., 2021; Jordan et al., 2022).

3.2. Merging Notation

In our context, we assume LoRA modules are merged via summation and is uniform across layers, but our results generalize to other merging functions as well. For simplicity, we define everything in terms of a single layer across multiple models. To merge N LoRA adapters, where subscript i indicates the ith LoRA module, MM and DM are defined:

$$L_{MM} = \sum_{i=1}^{N} \mathbf{B}_i * \mathbf{A}_i, L_{DM} = (\sum_i \mathbf{B}_i)(\sum_i \mathbf{A}_i) \quad (2)$$

3.3. Noise in Direct-Merge

Expanding L_{DM} from Equation (2) yields:

$$L_{DM} = (\sum_i \mathbf{B}_i)(\sum_i \mathbf{A}_i) = \sum_i (\mathbf{B}_i\mathbf{A}_i) + \sum_{i \neq j} \mathbf{B}_i\mathbf{A}_j \quad (3)$$

Let us also define $M = \sum_{i \neq j} \mathbf{B}_i\mathbf{A}_j$. Now, we rewrite (2) as $L_{DM} = L_{MM} + M$. Since M is composed of B_i and A_j terms that originate from different adapters, we hypothesize that their composition harms model performance. To test whether this interference term degrades model performance, we simulate M with gaussian noise and demonstrate that the decrease in performance due to the M is greater than or equal to the decrease resulting from the simulated noise.

3.4. Multiplied-Merge with SVD

Given the noise inherent to L_{DM}, it may be optimal to retain the performance benefits of L_{MM} and then find a low rank decomposition back into structure of L_{DM}. Here, we demonstrate that a simple SVD-based method can accomplish this. Let $SVD_r(\cdot) = U\Sigma_{[:r]}V^T$ indicate a function that takes the SVD of a matrix and retains only the r largest singular values. $SVD_r(L_{MM})$ then defines $\mathbf{A}$ as $\Sigma_{[:r]}V^T$ and $\mathbf{B}$ as U.

Mathematically, we will show that the magnitude of the error resulting from truncated SVD is bounded above by the mag-

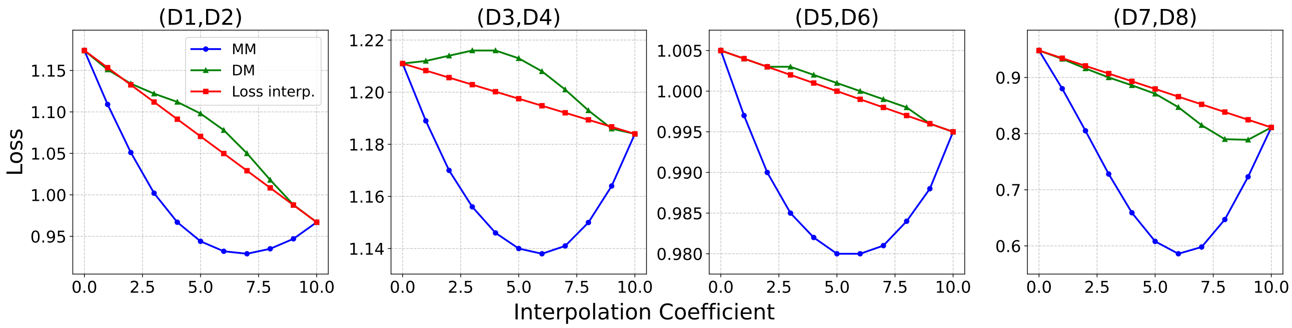

Figure 1. Linear Mode Connectivity Analysis (LMC) of L_{MM} and L_{DM} on all dataset pairs. Lines show loss at different interpolation values. L_{MM} consistently exhibits LMC (loss below interpolated loss line), while L_{DM} shows higher loss barriers.

nitude of M. Empirically, we then show that $SVD_r(L_{MM})$ outperforms the SOTA L_{DM} methods.

Math Proof Here, we show SVD truncation is a closer approximation to L_{MM} than L_{DM}. Definitionally:

$$SVD_r(L_{MM}) = SVD(L_{MM}) - SVD_{err}(L_{MM}) \quad (4)$$

where $SVD_{err}(L_{MM})$ is the error due to truncation. Specifically, $SVD(L_{MM}) = U\Sigma V^T$ and $SVD_{err}(L_{MM}) = U\Sigma_{[r+1:n]}V^T$ where $\Sigma_{[i:j]}$ indicates the ith through jth singular values.

The Eckart-Young-Mirsky theorem indicates that SVD provides the closest rank r approximation to any matrix. Since $SVD_r(L_{MM})$ and $LoRA_{DM}$ are both rank r, $SVD_r(L_{MM})$ is a closer approximation. Giving:

$$||L_{MM} - SVD_r(L_{MM})|| \leq ||L_{MM} - L_{DM}|| \quad (5)$$

Substituting Equation (2) (right) and Equation (4) gives:

$$\begin{aligned} &||L_{MM} - (SVD(L_{MM}) + SVD_{err}(L_{MM}))|| \leq \\ &||L_{MM} - (L_{MM} + M)|| \end{aligned} \quad (6)$$

$$||SVD_{err}(L_{MM})|| \leq ||M|| \quad (7)$$

Thus the error due to approximating L_{MM} with SVD is bounded above by the error due to using L_{DM}.

4. Results

4.1. Experimental Setup

We use openai/clip-vit-base-patch32 as our base model. We denote each dataset as D_i where $\{D_1$ = SVHN (Netzer et al., 2011), D_2 = GTSRB (Stallkamp et al., 2011), D_3 = DTD (Cimpoi et al., 2013), D_4 = RESISC45 (Cheng et al., 2017), D_5 = Stanford-Cars (Krause et al., 2013), D_6 = Sun397 (Xiao et al., 2014), D_7 = Eurosat (Helber et al., 2017), D_8 = MNIST (LeCun & Cortes, 2005)$\}$. We used the LoRA adapters (rank=16) and evaluation benchmarks from in (Tang et al., 2024). Out of the 8 available datasets, we randomly select 4 pairs of datasets (D_i, D_j) and fix them for each experiment. We also evaluate all eight datasets $D_1 - D_8$ merged together. For each dataset, we use the respective LoRA. For each experiment, we apply four different merging methods: Averaging, Task-Arithmetic (TA) Merging, TIES-merging, and DARE merging. For TA, TIES, DARE, we tune the hyper-parameters via a linear search on the eight combined datasets and fix them for all experiments.

4.2. Performance Comparison Between Multiplied Merging and Direct Merging

First, we compare the accuracy of L_{MM} and L_{DM} for each merging method across each pair of datasets. In Table 1, L_{DM} outperforms L_{MM} by $+0.09\%$ when using TA merging on dataset pair (D5, D6), but has worse performance in the remaining 19 comparisons. When merging 2 LoRA, L_{MM} outperforms L_{DM} by $+2.97\%$ on average. Notably, when merging 8 LoRA, L_{MM} achieves, on average, $+50.15\%$ accuracy compared to L_{DM} across all merging methods. This error in merging multiple LoRAs is more thoroughly investigated in section 4.4.

4.3. Equivalence of SVD-Based Approximation to Multiplied Merging

Since L_{DM} lags behind L_{MM}, we hypothesize that merging with L_{MM} and then reducing the rank with a truncated SVD can help mitigate this gap. So, we compare $SVD_{16}(L_{MM})$ with L_{MM}. We use $r = 16$ to match the rank of $LoRA_{DM}$. In Table 2, $SVD_{16}(\cdot)$ has a negligible performance drop compared to L_{MM}. L_{MM} is on average $+0.46\%$ across all methods on paired LoRAs, while $SVD_{16}(\cdot)$ is $+0.52\%$ compared to L_{MM} when merging all 8 LoRAs.

Dataset (→)	(D1, D2)			(D3, D4)			(D5, D6)			(D7, D8)			Merge All		
Method (↓)	**MM**	**DM**	**Noise**	**MM**	**DM**	**Noise**	**MM**	**DM**	**Noise**	**MM**	**DM**	**Noise**	**MM**	**DM**	**Noise**
Averaging	82.00	76.50	81.91	68.30	65.64	68.27	69.60	68.83	69.62	93.90	89.33	93.99	64.40	61.31	64.10
Task Arithmetic	85.50	81.90	85.89	72.20	71.43	72.28	70.60	70.69	70.57	96.40	96.00	96.38	73.30	6.20	71.50
TIES	81.40	73.51	81.54	67.40	61.90	67.26	69.60	66.46	69.64	94.30	87.32	93.99	68.50	5.60	67.50
DARE	86.00	81.74	85.83	72.40	71.23	72.24	70.60	70.73	70.61	96.40	95.94	96.38	73.50	6.00	70.80
Average	83.73	78.41	83.79	70.08	67.55	70.01	70.10	69.18	70.11	95.25	92.15	95.19	69.93	19.78	68.50

Table 1. Accuracy comparison between Model Merging (MM), Direct Merging (DM), and Noise Simulation

Dataset (→)	(D1, D2)		(D3, D4)		(D5, D6)		(D7, D8)		Merge All	
Method (↓)	**MM**	**SVD**	**MM**	**SVD**	**MM**	**SVD**	**MM**	**SVD**	**MM**	**SVD**
Averaging	82.00	81.80	68.30	67.90	69.60	69.60	93.90	93.90	64.40	64.10
Task Arithmetic	85.50	85.74	72.20	71.82	70.60	70.47	96.40	96.3	73.30	72.23
TIES	81.40	78.10	67.40	66.06	69.60	69.27	94.30	90.61	68.50	73.23
DARE	86.00	85.68	72.40	71.99	70.60	70.49	96.40	96.30	73.50	72.20
Average	83.73	82.83	70.08	70.00	70.10	70.08	95.25	94.40	69.93	70.44

Table 2. Accuracy comparison between of $LoRA_{MM}$ vs $SVD_{16}(LoRA_{MM})$.

Rank (↓)	Method (↓)	(D1, D2)	(D3, D4)	(D5, D6)	(D7, D8)	All	Average
8	Lora Lego	73.90	65.10	68.20	82.50	61.10	70.16
	MM SVD	81.40	67.20	69.40	93.70	63.70	77.90
16	Lora Lego	77.90	64.90	68.40	82.30	61.90	71.08
	MM SVD	81.80	67.90	69.60	93.90	64.10	78.30
32	Lora Lego	76.40	64.70	68.60	87.60	63.3	72.12
	MM SVD	82.00	68.30	69.60	93.90	64.30	78.50

Table 3. Accuracy comparison between $SVD_r(LoRA_{MM})$ (LoRA averaging) and LoRA LEGO.

4.4. Analysis

SVD-Based Approach Outperforms State-of-the-Art Methods. Next, we compare $SVD_r(L_{MM})$ with a SOTA $LoRA_{DM}$ merging method. LoRA LEGO (Zhao et al., 2024) is a method that decomposes LoRA modules into rank-1 units and then clusters them together. The centroid of each cluster contributes 1 rank to the final merged-LoRA, thereby giving users fine-grain control of the merged rank. However, LoRA LEGO introduces the same noise present in $LoRA_{DM}$ as LoRA units from different adapters are averaged together. In our comparison, we select r = [8, 16, 32] and use simple averaging to compute L_{MM}. As shown in Table 3, for rank 8, $SVD_r(L_{MM})$ achieves $+7.74\%$ accuracy compare to Lego LoRA, $+7.22\%$ at rank 16, and $+6.38\%$ at rank 32. This indicates that $SVD_r(L_{MM})$ is able to outperform SOTA L_{DM} methods, while retaining the ability to dynamically select merged LoRA rank.

Linear Mode Connectivity Analysis We demonstrate that L_{MM} is LMC while L_{DM} is not. For each task, we interpolated the pairs of LoRAs with coefficients $a \in [0.1, 0.2, ...0.9, 1.0]$, and calculate the average cross entropy loss on the two test sets. We then plot the interpolated losses and the loss of the interpolated models. Models are LMC when the interpolated model's loss is less than the interpolated loss for all interpolation values. Fig. 1 shows that L_{MM} is LMC on all datasets, whereas L_{DM} is only LMC on a single dataset.

Interference Impact and Scaling Effects As shown in section 3.3, $L_{DM} = L_{MM} + M$. To simulate the impact of M, we sample a noise matrix $N \in \mathbb{R}^{n\times n}$ with $N_{ij} \sim \mathcal{N}(mean(M), std(M))$ and demonstrate that $L_{MM} + N$ performs better than or equal to L_{DM}, indicating that the term M is source of degradation in merging performance. In Table 1, adding N to L_{MM} has a minimal effect on accuracy (-0.025% difference on average), however, when merging on eight LoRA, this noise actually does better by $+0.143\%$.

5. Conclusion

We demonstrated that merging LoRA with Direct-Merge can maintain a memory-efficient low-rank structure but introduces interference terms that degrade model performance and break LMC. By using truncated SVD on top of Mutliplied-Merge, we show that it is easy to retain memory-efficient structure with virtually no performance cost.

References

Achiam, O. J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., and et al., D. A. Gpt-4 technical report. 2023. URL https://api.semanticscholar.org/CorpusID:257532815.

Cheng, G., Han, J., and Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105:1865–1883, 2017. URL https://api.semanticscholar.org/CorpusID:3046524.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2013. URL https://api.semanticscholar.org/CorpusID:4309276.

Daheim, N., Möllenhoff, T., Ponti, E., Gurevych, I., and Khan, M. E. Model merging by uncertainty-based gradient matching. *ArXiv*, abs/2310.12808, 2023. URL https://api.semanticscholar.org/CorpusID:264306115.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314, 2023. URL https://api.semanticscholar.org/CorpusID:258841328.

Entezari, R., Sedghi, H., Saukh, O., and Neyshabur, B. The role of permutation invariance in linear mode connectivity of neural networks. *ArXiv*, abs/2110.06296, 2021. URL https://api.semanticscholar.org/CorpusID:238743980.

Feng, S., Wang, Z., Wang, Y., Ebrahimi, S., Palangi, H., Miculicich, L., Kulshrestha, A., Rauschmayr, N., Choi, Y., Tsvetkov, Y., Lee, C.-Y., and Pfister, T. Model swarms: Collaborative search to adapt llm experts via swarm intelligence. *ArXiv*, abs/2410.11163, 2024. URL https://api.semanticscholar.org/CorpusID:273350735.

Frankle, J., Dziugaite, G. K., Roy, D. M., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:209324341.

Helber, P., Bischke, B., Dengel, A. R., and Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12:2217–2226, 2017. URL https://api.semanticscholar.org/CorpusID:11810992.

Hu, J. E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. URL https://api.semanticscholar.org/CorpusID:235458009.

Huang, C., Liu, Q., Lin, B. Y., Pang, T., Du, C., and Lin, M. Lorahub: Efficient cross-task generalization via dynamic lora composition. *ArXiv*, abs/2307.13269, 2023. URL https://api.semanticscholar.org/CorpusID:260155012.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. *ArXiv*, abs/2212.04089, 2022. URL https://api.semanticscholar.org/CorpusID:254408495.

Jordan, K., Sedghi, H., Saukh, O., Entezari, R., and Neyshabur, B. Repair: Renormalizing permuted activations for interpolation repair. *ArXiv*, abs/2211.08403, 2022. URL https://api.semanticscholar.org/CorpusID:253523197.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013. URL https://api.semanticscholar.org/CorpusID:14342571.

LeCun, Y. and Cortes, C. The mnist database of handwritten digits. 2005. URL https://api.semanticscholar.org/CorpusID:60282629.

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024.

Lu, Z., Fan, C., Wei, W., Qu, X., Chen, D., and Cheng, Y. Twin-merging: Dynamic integration of modular expertise in model merging. *ArXiv*, abs/2406.15479, 2024. URL https://api.semanticscholar.org/CorpusID:270702345.

Matena, M. and Raffel, C. Merging models with fisher-weighted averaging. *ArXiv*, abs/2111.09832, 2021. URL https://api.semanticscholar.org/CorpusID:244345933.

Mavromatis, C., Karypis, P., and Karypis, G. Pack of llms: Model fusion at test-time via perplexity optimization. *ArXiv*, abs/2404.11531, 2024. URL https://api.semanticscholar.org/CorpusID:269188153.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Reading digits in natural images with unsupervised feature learning. 2011. URL https://api.semanticscholar.org/CorpusID:16852518.

Prabhakar, A., Li, Y., Narasimhan, K., Kakade, S. M., Malach, E., and Jelassi, S. Lora soups: Merging loras for practical skill composition tasks. *ArXiv*, abs/2410.13025, 2024. URL https://api.semanticscholar.org/CorpusID:273404154.

Reid, M., Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T. P., Alayrac, J.-B., and nd et al, R. S. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *ArXiv*, abs/2403.05530, 2024. URL https://api.semanticscholar.org/CorpusID:268297180.

Shah, V., Ruiz, N., Cole, F., Lu, E., Lazebnik, S., Li, Y., and Jampani, V. Ziplora: Any subject in any style by effectively merging loras. *ArXiv*, abs/2311.13600, 2023. URL https://api.semanticscholar.org/CorpusID:265351656.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. The german traffic sign recognition benchmark: A multi-class classification competition. *The 2011 International Joint Conference on Neural Networks*, pp. 1453–1460, 2011. URL https://api.semanticscholar.org/CorpusID:15926837.

Stoica, G., Ramesh, P., Ecsedi, B., Choshen, L., and Hoffman, J. Model merging with svd to tie the knots. *ArXiv*, abs/2410.19735, 2024. URL https://api.semanticscholar.org/CorpusID:273638541.

Tam, D., Bansal, M., and Raffel, C. Merging by matching models in task parameter subspaces. *Trans. Mach. Learn. Res.*, 2024, 2023. URL https://api.semanticscholar.org/CorpusID:266053657.

Tang, A. Q., Shen, L., Luo, Y., Hu, H., Du, B., and Tao, D. Fusionbench: A comprehensive benchmark of deep model fusion. *ArXiv*, abs/2406.03280, 2024. URL https://api.semanticscholar.org/CorpusID:270257718.

Touvron, H., Martin, L., Stone, K. R., Albert, P., Almahairi, A., Babaei, Y., lay Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D. M., Blecher, L., and et al. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023. URL https://api.semanticscholar.org/CorpusID:259950998.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Huggingface's transformers: State-of-the-art natural language processing, 2019. https://arxiv.org/abs/1910.03771.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., 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. *ArXiv*, abs/2203.05482, 2022. URL https://api.semanticscholar.org/CorpusID:247362886.

Wu, X., Huang, S., and Wei, F. Mixture of lora experts. *ArXiv*, abs/2404.13628, 2024. URL https://api.semanticscholar.org/CorpusID:269293160.

Xiao, J., Ehinger, K. A., Hays, J., Torralba, A., and Oliva, A. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119:3–22, 2014. URL https://api.semanticscholar.org/CorpusID:10224573.

Yadav, P., Choshen, L., Raffel, C., and Bansal, M. Compeft: Compression for communicating parameter efficient updates via sparsification and quantization. *ArXiv*, abs/2311.13171, 2023a. URL https://api.semanticscholar.org/CorpusID:265351803.

Yadav, P., Tam, D., Choshen, L., Raffel, C., and Bansal, M. Ties-merging: Resolving interference when merging models. In *Neural Information Processing Systems*, 2023b. URL https://api.semanticscholar.org/CorpusID:259064039.

Yang, E., Wang, Z., Shen, L., Liu, S., Guo, G., Wang, X., and Tao, D. Adamerging: Adaptive model merging for multi-task learning. *ArXiv*, abs/2310.02575, 2023. URL https://api.semanticscholar.org/CorpusID:263620126.

Yu, L., Bowen, Y., Yu, H., Huang, F., and Li, Y. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *ArXiv*, abs/2311.03099, 2023. URL https://api.semanticscholar.org/CorpusID:265034087.

Zhao, Z., Shen, T., Zhu, D., Li, Z., Su, J., Wang, X., Kuang, K., and Wu, F. Merging loras like playing lego: Pushing the modularity of lora to extremes through rank-wise clustering. *ArXiv*, abs/2409.16167, 2024. URL https://api.semanticscholar.org/CorpusID:272831995.