STAR: Spectral Truncation and Rescale for Model Merging

Yu-Ang Lee[1,2][*], Ching-Yun Ko[2], Tejaswini Pedapati[2],
I-Hsin Chung[2], Mi-Yen Yeh[3], Pin-Yu Chen[2]

[1] Data Science Degree Program, National Taiwan University and Academia Sinica,
[2]IBM Research, [3]Academia Sinica
r12946015@ntu.edu.tw, cyko@ibm.com, tejaswinip@us.ibm.com
ihchung@us.ibm.com, miyen@iis.sinica.edu.tw, pin-yu.chen@ibm.com

Abstract

Model merging is an efficient way of obtaining a multi-task model from several pretrained models without further fine-tuning, and it has gained attention in various domains, including natural language processing (NLP). Despite the efficiency, a key challenge in model merging is the seemingly inevitable decrease in task performance as the number of models increases. In this paper, we propose **S**pectral **T**runcation **A**nd **R**escale (STAR) that aims at mitigating "merging conflicts" by truncating small components in the respective spectral spaces, which is followed by an automatic parameter rescaling scheme to retain the nuclear norm of the original matrix. STAR requires no additional inference on original training data and is robust to hyperparamater choice. We demonstrate the effectiveness of STAR through extensive model merging cases on diverse NLP tasks. Specifically, STAR works robustly across varying model sizes, and can outperform baselines by 4.2% when merging 12 models on Flan-T5. Our code is publicly available at this https URL.

1 Introduction

With the popularity of pretrained models on large neural networks, the same architecture is often deployed to fine-tune individual natural language processing (NLP) tasks. A natural question then arises about whether it is possible to merge these same-architecture fine-tuned models into one multi-task model. For example, researchers are interested in understanding if we can empower a fine-tuned conversational large language model (LLM) with reasoning capabilities by merging with an LLM specializing in solving math problems. Specifically, Ilharco et al. (2022) has formally defined a *task vector* as $\theta_{ft} - \theta_{pre}$, where θ_{pre} and θ_{ft} denote the vectorized parameters of the pre-trained model and the fine-tuned model, respectively. Thus, task vectors
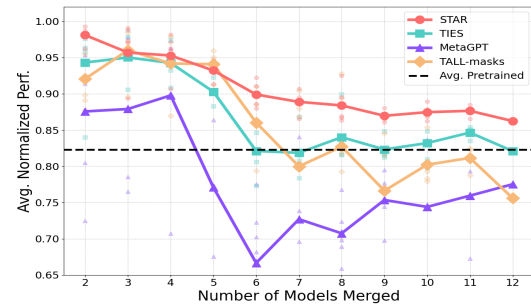

Figure 1: The averaged normalized performance of Flan-T5-base merged models by TIES (Yadav et al., 2024), MetaGPT (Zhou et al., 2024), TALL-masks (Wang et al., 2024), and STAR (this paper).

mark the updates made to the pretrained model's weights when fine-tuned on specific tasks. Then, *model merging* essentially studies ways of fusing different task vectors that are trained separately and merging them with the pretrained model. However, as the number of fine-tuned models increases, the multi-task performance of their merged model also decreases drastically. Fig. 1 shows the averaged normalized performance (y-axis) v.s. the number of models merged (x-axis). Furthermore, we point out that when the number of models exceeds a certain threshold, the multi-task performance of the merged model could be even worse than that of the original pretrained model, diminishing the fundamental goal of model merging. For example, TIES (Yadav et al., 2024), MetaGPT (Zhou et al., 2024), and TALL-masks (Wang et al., 2024) merged models drop below 0.82 when we merge 6, 5, and 7 fine-tuned models, respectively, in Fig. 1.

The complexity of existing model merging methods varies largely depending on whether they require fine-tuning or inference on training data (Yang et al., 2024). In this paper, we study the "data-free" setting when we are not authorized to change the fine-tuning protocol nor do we have access to the training data. In this work, we propose to use spectral decomposition (e.g. singular value
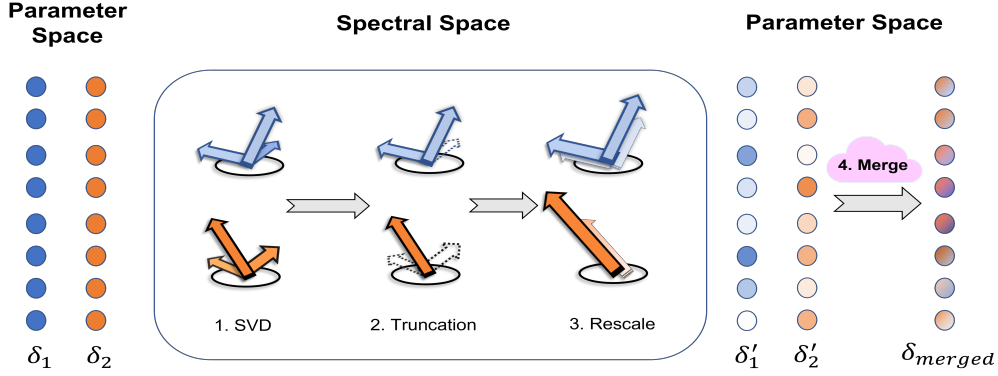

Figure 2: An overview of the **STAR workflow**. When merging two task vectors, δ_1 and δ_2, (1) STAR transforms both task vectors into their spectral spaces with their singular vectors being the orthogonal basis using singular value decomposition (SVD) (singular values are represented by the length of the arrows), (2) STAR removes redundant dimensions by truncating singular vectors with small singular values, (3) STAR restores the original nuclear norm by rescaling the truncated SVD, and (4) STAR reconstructs the parameters by multiplying components back to form the weight matrices and then perform simple averaging.

decomposition, SVD) to remove noisy components on model merging. We will also motivate the potential gain of our spectral space merging scheme by comparing the upper bounds of the task conflicts. A rescaling step is then followed to restore the original nuclear norm. We give the overview of the proposed method in Fig. 2. Our proposed merging scheme, **S**pectral **T**runcation **A**nd **R**escale (STAR), is effective and efficient as it requires no additional inference on original training data and is not sensitive to hyperparameters. Our extensive experimental results show that STAR is superior across various model size settings and can effectively merge up to 20 models while achieving positive performance gains, compared to the pretrained model before merging.

2 Background and Related Work

2.1 Notations and Problem Definition

We denote the weight matrices of a pretrained LM by θ_{pre}^l for $l = \{1, \ldots, L\}$, where L is the total number of such matrices. Let θ_{pre} denote the concatenation of all vectorized weight matrices and θ_{ft} denote the updated model parameters after fine-tuning on task $\mathcal{T}$. A task vector δ is then defined as the difference between θ_{ft} and θ_{pre}, i.e., $\delta = \theta_{\text{ft}} - \theta_{\text{pre}}$ (Ilharco et al., 2022). Given T fine-tuned models, model merging fuses $\{\delta_1, \ldots, \delta_T\}$ into a merged δ_{merged} such that $\theta_{\text{pre}} + \delta_{\text{merged}}$ still performs well on T tasks simultaneously.

2.2 Related Work

Model merging methods belong to two categories: Pre-merging and During-merging methods (Yang et al., 2024). While pre-merging methods focus on renovating the fine-tuning step such that the fine-tuned models suit model merging better (Ortiz-Jimenez et al., 2024; Imfeld et al., 2023; Guerrero Pena et al., 2022), during-merging methods assume no access to the fine-tuning and work directly on models given. Recently, Yang et al. (2024) further classifies during-merging methods into five sub-classes, of which STAR is most related to the weighted-based and subspace-based methods.

Weighted-based. As base merging methods such as Ilharco et al. (2022) applies the same scaling across all model layers and tasks, weighted-based methods take the importance of parameters into account and scale differently, e.g. Matena and Raffel (2022); Tam et al. (2024) leverage Fisher matrix for assessing the importance of parameters, while others utilize Hessian estimation or entropy, etc (Daheim et al., 2023; Yang et al., 2023). However, these methods require inference through original data, making it infeasible with limited compute or access to task data. MetaGPT (Zhou et al., 2024) proposes a closed form solution for scaling task vectors by minimizing the average loss of the merged model and the independent model.

Subspace-Based. Another line of work transforms task vectors into sparse subspaces (Davari and Belilovsky, 2023; Yadav et al., 2024; Wang et al., 2024; Huang et al., 2024), e.g. TIES (Yadav et al., 2024) trims task vectors to keep only the top $K\%$ parameters with the highest magnitude, before undergoing an elect-sign step to reduce sign conflicts; TALL-masks (Wang et al., 2024) constructs per-task masks that identifies important parameters within each task, which are then merged into one general mask based on consensus among multiple

per-task masks.

STAR differs from the above as it transforms task vectors to the spectral spaces, and its truncation and scale are task-dependent and layer-specific.

3 Methodology

Sec. 3.1 provides the rationale behind performing truncations in the spectral space. Sec. 3.2 defines the rescaling step for restoring the nuclear norm. Sec. 3.3 gives the complete STAR algorithm.

3.1 Spectral Truncation

Let $\mathcal{T}_1, \mathcal{T}_2$ be two fine-tuning tasks that yield task vectors δ_{T_1} and δ_{T_2}. Take the entries correspond to a weight matrix and reconstruct them into A, B from δ_{T_1} and δ_{T_2}, respectively. Suppose A and B admit SVD into $\sum_i \sigma_i^A u_i^A (v_i^A)^T$ and $\sum_i \sigma_i^B u_i^B (v_i^B)^T$, one can obtain the matrix rank by the number of nonzero singular values. By selecting only the top few singular values and vectors (i.e. truncated SVD), we naturally find the principal components and remove the redundant dimensions, effectively reducing the rank of the matrix. As small singular values often correlate with noise or fine details, low-rank prior is also widely used in compressed sensing and denoising applications in signal processing (Dabov et al., 2007; Candes and Plan, 2010; Cai et al., 2010; Candes and Recht, 2012).

Besides extracting principal components, we also give a high-level illustration of why using truncated SVD on A and B separately can help reduce conflicts during model merging. Assume $\mathcal{T}_1$ is associated with data manifold $\mathcal{D}_\mathcal{A}$. For $x \in \mathcal{D}_\mathcal{A}$, we essentially hope $(A \oplus B)x$ to be close to Ax while excelling at $\mathcal{T}_2$ after merging, where $\oplus$ denotes the merging operation. Let us consider the merging operation to be plainly $A + B$, then the level of conflicts can be measured by $\|Bx\|$. By expressing $x \in \mathcal{D}_\mathcal{A}$ via the right singular vectors of A, $x = \sum_j \alpha_j v_j^A$, we prove in Sec. A.1 that we have $\|Bx\| \leq r^B \beta \sqrt{r^A}$, where $\beta = \max_{i,j} |\sigma_i^B \alpha_j|$, and r^A and r^B are the original ranks of A and B. By truncating B to rank-r, this upper bound is lowered by $(r^B - r)\beta\sqrt{r^A}$, implying potentially less conflicts in model merging.

3.2 Rescale to Restore Matrix Nuclear Norm

As model merging favors spectral truncation as discussed in Sec. 3.1, a caveat is the resulting change in the ratio between the pretrained model

Figure 3: An example of the automatic rank determination by STAR ($\eta = 40$) on PIQA's task vector with Flan-T5-large.

and the task vector. Roughly, one sees that $\|Ax\| = \|\sum_i \sigma_i^A u_i^A (v_i^A)^T \sum_j \alpha_j v_j^A\| = \|\sum_i \sigma_i^A \alpha_i u_i^A\|$ and can at most be $\sum_{i=r+1} \|\sigma_i^A \alpha_i\|$ smaller with the truncated A. Therefore, the performance on the fine-tuning task $\mathcal{T}_1$ might be compromised. On that account, it is crucial to include a step where we rescale the spectral-truncated weight matrices back to their original "size", similar to the compensation operation in dropout. We propose to retain matrix nuclear norm (aka Schatten 1-norm or trace norm) as it is a proper measure of matrix "size", especially in low-rank approximation contexts as nuclear norm is a convex relaxation of the rank function (Candes and Recht, 2012). Specifically, we rescale the remaining singular values by

$$\sigma_k' = \frac{\sum_i \sigma_i}{\sum_{i=1}^r \sigma_i} \cdot \sigma_k, \quad \forall k \in [1, r].$$

3.3 STAR: Spectral Truncate And Rescale

Now that we have elaborated on the two key components of STAR, we explain the complete workflow in the following. With T task vectors, we transform them into respective spectral spaces via SVD, and their ranks are determined by $r = \arg\min_k \left(\frac{\sum_{i=1}^k \sigma_i}{\sum_i \sigma_i} \geq \eta\% \right)$, where η is a tunable parameter. Then, we follow Section 3.2 to rescale back to their original nuclear norm. Finally, STAR reconstructs T task vectors from their decompositions and perform simple averaging to obtain δ_{merged}. We give the full STAR model merging algorithm in Alg. 1 in appendix.

We note that as the distribution of singular values varies both within and across task vectors, truncating components adaptively allows different ranks across not only tasks and even layers (e.g. Fig. 3).
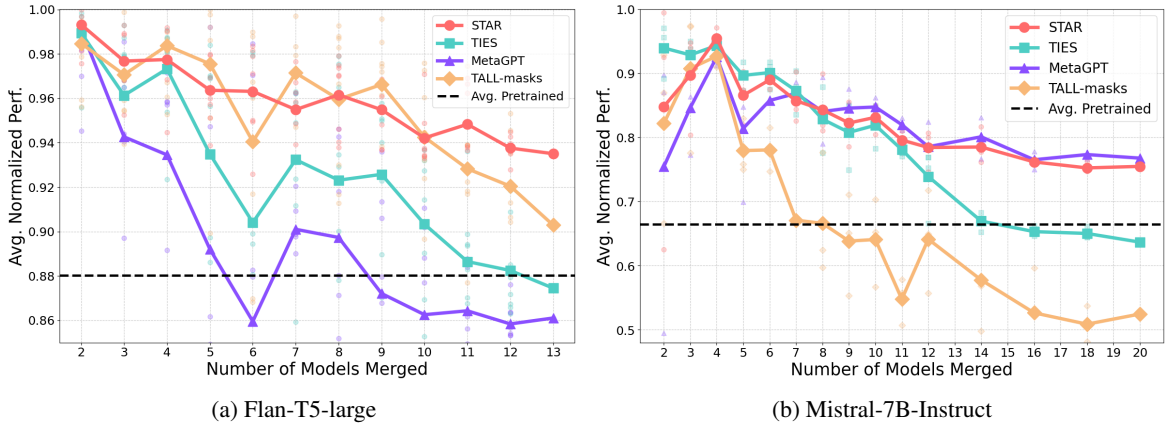
498

(a) Flan-T5-large

(b) Mistral-7B-Instruct

Figure 4: Model merging results on Flan-T5-large and Mistral-7B-Instruct. For all numbers of models merged, we sampled 5 task combinations for Flan-T5 and 3 for Mistral, with the sampled combinations represented by shaded dots and the average depicted by solid lines. While STAR remains a strong model merging method, TIES, TALL-masks and MetaGPT can be more sensitive to model architecture choice.

4 Experiments

4.1 Experimental Setup

Models. We consider both encoder-decoder models (e.g. Flan-T5-base/large) (Chung et al., 2024) and decoder-only model (e.g. Mistral-7B-Instruct-v0.2) (Jiang et al., 2023). For Flan-T5-base/large, we use finetuned models on GLUE from Fusion-Bench (Tang et al., 2024), together with additional fine-tuned models on Finance (Malo et al., 2014), IMDB (Maas et al., 2011), AG News (Zhang et al., 2015), BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), and HellaSwag (Zellers et al., 2019) by ourselves, bringing the total number of task vectors to 13. For Mistral-Instruct, we randomly select 20 models directly from the Lots of LoRAs collection (Brüel-Gabrielsson et al., 2024), which covers a range of NLI tasks. All models considered herein are LoRA finetuned (Hu et al., 2021) with rank 16 and scaling factor (alpha) set to 32. Details about the models are in Appendix Sec. A.6. To understand how each merging method performs on n models, we randomly sample n tasks and report their average results.

Hyperparameters. Without otherwise specified, we let $K = 20$ for TIES (the default parameter in (Yadav et al., 2024)), $\lambda_t = 0.4$ for TALL-masks (the middle value searched by (Wang et al., 2024)), and $\eta = 40$ for STAR.

Evaluation metric. Following Tang et al. (2024); Brüel-Gabrielsson et al. (2024), performances on QASC (Khot et al., 2020) and STSB (Cer et al., 2017) are evaluated by F1 score and Spearman's coefficient, respectively, and accuracy for all other tasks. If the correct output appears within the first
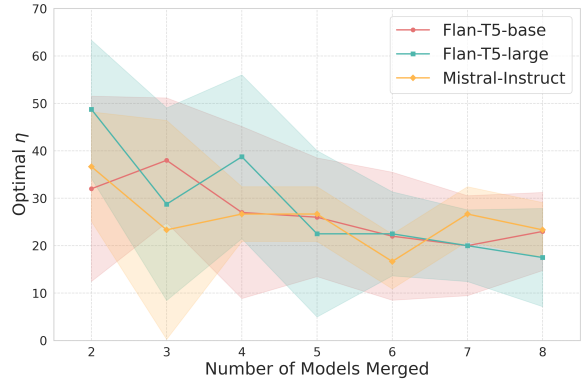

Figure 5: The mean and standard deviation of the optimal η, which yields the best merged model performance, decrease as the number of merged models increases.

10 tokens generated by the merged model, the response is deemed correct. For a model merged on t tasks, we report the normalized average performance (Ilharco et al., 2022; Yadav et al., 2024) defined by $\frac{1}{t} \sum_i^t \frac{(\text{Merged Model Perf.})_i}{(\text{Finetuned Model Perf.})_i}$. We further measure the performance of the pretrained model by $\frac{1}{T} \sum_{i=1}^T \frac{\text{Pretrained Model Perf.}_i}{\text{Finetuned Model Perf.}_i}$. If the merged model performs worse than the pretrained model, then model merging loses its purpose.

4.2 Performance Comparison

We compare STAR to other data-free approaches, including TIES (Yadav et al., 2024), TALL-masks (Wang et al., 2024), which we apply on top of Task Arithmetic (Ilharco et al., 2022), i.e., Consensus Task Arithmetic (without tuning the data-dependent hyperparameter λ_t), and MetaGPT (Zhou et al., 2024). Due to the page limit, we defer the discussion around EMR-Merging (Huang et al., 2024) and DARE (Yu et al.,

Rank Kept	Rescale	MRPC	Finance	HellaSwag	PIQA	Avg. Normalized
r=2	No	73.36	91.19	77.75	**80.75**	97.17
	Yes	**74.05**	**96.04**	**79.40**	80.25	**99.01**
r=4	No	73.27	94.71	78.35	**81.00**	98.32
	Yes	**73.79**	**96.04**	**79.20**	80.75	**99.02**
r=8	No	73.44	94.71	78.70	**81.00**	98.48
	Yes	73.44	**95.59**	**78.80**	80.50	**98.58**
r=12	No	73.44	94.71	78.55	**81.00**	98.44
	Yes	73.44	**95.15**	**78.85**	81.25	**98.72**

Table 1: The ablation study of the rescaling step to restore nuclear norms (i.e. Sec. 3.2).

(a) Flan-T5-base

(b) Flan-T5-large

Figure 6: The average model merging results on Flan-T5-base and Flan-T5-large over a range of possible hyperparameter choices.

2024) to appendix Sec. A.3 and Sec. A.4.

The results on Flan-T5-large and Mistral-7B-Instruct are shown in Fig. 4 and Flan-T5-base in Fig. 1. We note that similar trends as Fig. 1 can be seen in Fig. 4 where the averaged normalized performance decreases as the number of models merged increases, with STAR's performance decay being the slowest across models. On Flan-T5-base, MetaGPT tends to fail quickly, echoing with the findings in (Zhou et al., 2024) - MetaGPT may face limitations when merging models of smaller sizes (e.g. Flan-T5-base has only 0.25B parameters) due to its reliance on NTK linearization. To examine the full potential of each algorithm, we also perform grid search for TIES and STAR and report the best result in Appendix Sec. A.5.
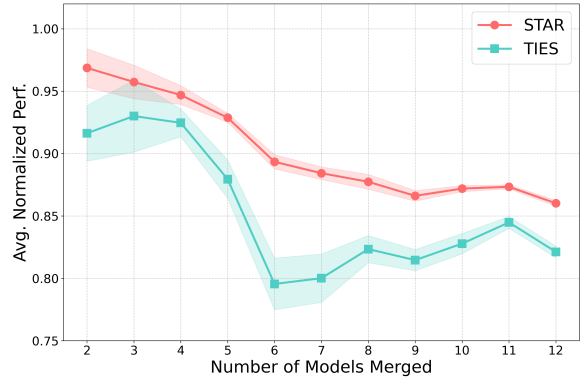
4.3 Additional Results

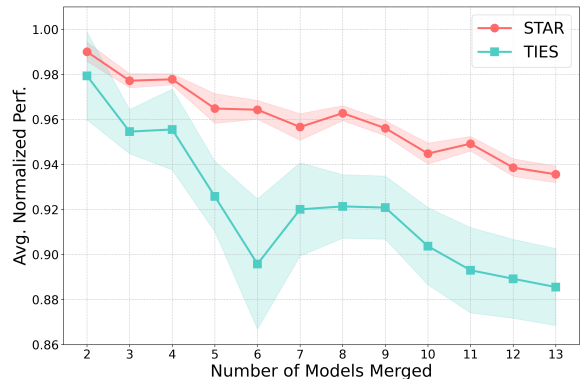**Ablation studies on restoring the nuclear norm** In Table 1, we give an example of merging 4 fine-tuned Flan-T5-large models with and without rescale to restore the matrix nuclear norm. We see that rescale is crucial especially when we use low-rank approximations (e.g. rank-2).

Sensitivity analysis of η. As η is the only tunable hyperparameter in STAR, we further show in Fig. 6 that η is robust across different model merging combinations and numbers of models merged, compared to the baseline (e.g. TIES). Specifically, we allow STAR to choose η from $\{10, 20, \ldots, 70\}$ and TIES to choose K from $\{1, 5, 10, 20, \ldots, 70\}$. From the standard deviation in Fig. 6, it can indeed be seen that STAR is not sensitive to η, sparing users' need to fine-tune η during the deployment.

Optimal η varies as number of models merged. Following Ilharco et al. (2022), we report the optimal η when merging different number of models in Fig. 5. By searching for η within $\{10, 20, \ldots, 70\}$ across all sampled model merging combinations, we observed an interesting trend: as the number of merged models increases, the optimal η gradually decreases, indicating that higher truncation for each task vector is necessary.

5 Conclusion

In this paper, we propose Spectral Truncation And Rescale (STAR) for model merging by removing noisy components via spectral decomposition and restoring the original nuclear norm through rescaling. STAR requires no additional inference and is robust to different hyperparameter choices and language models. STAR provides a principaled way of automatic rank determination and is intuitively complimentary to other merging methods.

Limitation

While STAR demonstrates strong potential for practical model merging use cases across domains, its performance has been tested primarily on parameter-efficient fine-tuned (PEFT) models in

NLP. Additionally, STAR requires SVD to orthogonalize task vectors, which may introduce additional computational cost. However, users can mitigate this by leveraging fast SVD algorithms in the implementation.

Acknowledgement

References

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Rickard Brüel-Gabrielsson, Jiacheng Zhu, Onkar Bhardwaj, Leshem Choshen, Kristjan Greenewald, Mikhail Yurochkin, and Justin Solomon. 2024. Compress then serve: Serving thousands of lora adapters with little overhead. *arXiv preprint arXiv:2407.00066*.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982.

Emmanuel Candes and Benjamin Recht. 2012. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119.

Emmanuel J Candes and Yaniv Plan. 2010. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. 2007. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095.

Nico Daheim, Thomas Möllenhoff, Edoardo Maria Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. 2023. Model merging by uncertainty-based gradient matching. *arXiv preprint arXiv:2310.12808*.

MohammadReza Davari and Eugene Belilovsky. 2023. Model breadcrumbs: Scaling multi-task model merging with sparse masks. *arXiv preprint arXiv:2312.06795*.

Fidel A Guerrero Pena, Heitor R Medeiros, Thomas Dubail, Masih Aminbeidokhti, Eric Granger, and Marco Pedersoli. 2022. Re-basin via implicit sinkhorn differentiation. in 2023 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20237–20246.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. 2024. Emr-merging: Tuning-free high-performance model merging. *arXiv preprint arXiv:2405.17461*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Moritz Imfeld, Jacopo Graldi, Marco Giordano, Thomas Hofmann, Sotiris Anagnostidis, and Sidak Pal Singh. 2023. Transformer fusion with optimal transport. *arXiv preprint arXiv:2310.05719*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. *arXiv:1910.11473v2*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65.

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

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. 2024. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36.

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

Anke Tang, Li Shen, Yong Luo, Han Hu, Bo Do, and Dacheng Tao. 2024. Fusionbench: A comprehensive benchmark of deep model fusion. *arXiv preprint arXiv:2406.03280*.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. 2024. Localizing task information for improved model merging and compression. In *Forty-first International Conference on Machine Learning*.

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

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2023. Adamerging: Adaptive model merging for multi-task learning. *arXiv preprint arXiv:2310.02575*.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yuyan Zhou, Liang Song, Bingning Wang, and Weipeng Chen. 2024. Metagpt: Merging large language models using model exclusive task arithmetic. *arXiv preprint arXiv:2406.11385*.

A Appendix

A.1 Bounding $\|Bx\|$

Let r^A and r^B be the original ranks of A and B, $B = \sum_{i=1}^{r^B} \sigma_i^B u_i^B (v_i^B)^T$, $x = \sum_{j=1}^{r^A} \alpha_j v_j^A$, and $\{v_i^A\}_{i=1}^{r^A}$ and $\{v_i^B\}_{i=1}^{r^B}$ are orthonormal vectors, then we have

$$\|Bx\| = \|\sum_i \sigma_i^B u_i^B (v_i^B)^T \sum_j \alpha_j v_j^A\|$$

$$\leq \sum_i \|u_i^B\| \cdot |\sum_j \sigma_i^B \alpha_j (v_i^B)^T v_j^A|$$

$$\leq \sum_i \beta \cdot |\sum_j (v_i^B)^T v_j^A|$$

$$\leq \sum_{i=1}^{r^B} \beta \sqrt{r^A} \left(\sum_{j=1}^{r^A} \left((v_i^B)^T v_j^A \right)^2 \right)^{1/2} \tag{1}$$

$$= \sum_{i=1}^{r^B} \beta \sqrt{r^A} \left(\sum_{j=1}^{r^A} \langle v_i^B, v_j^A \rangle^2 \right)^{1/2}, \tag{2}$$

where $\beta = \max_{i,j} |\sigma_i^B \alpha_j|$, and inequality (1) uses Cauchy-Schwarz inequality. Then we show that

$$1 = \|v_i^B\|^2$$

$$= \| \sum_{j=1}^{r^A} \langle v_i^B, v_j^A \rangle v_j^A + v_i^{B\perp A} \|^2 \tag{3}$$

$$= \sum_{j=1}^{r^A} \| \langle v_i^B, v_j^A \rangle v_j^A \|^2 + \|v_i^{B\perp A}\|^2 \tag{4}$$

$$= \sum_{j=1}^{r^A} \langle v_i^B, v_j^A \rangle^2 + \|v_i^{B\perp A}\|^2$$

$$\geq \sum_{j=1}^{r^A} \langle v_i^B, v_j^A \rangle^2, \tag{5}$$

where equation (3) expresses v_i^B by $\{v_i^A\}_{i=1}^{r^A}$, and $v_i^{B\perp A}$ denotes the part of v_i^B that is orthogonal to the span of $\{v_i^A\}_{i=1}^{r^A}$. Equation (4) follows Pythagorean identity since $v_1^A, v_2^A, \ldots, v_{r_A}^A, v_i^{B\perp A}$ are pairwise-orthogonal vectors. Finally, with Equation (2) and (5), we have

$$\|Bx\| \leq r^B \beta \sqrt{r^A}.$$

A.2 Algorithm

Algorithm 1 Model merging by STAR

Input: $\boldsymbol{\theta}_{\text{pre}}, \{\boldsymbol{\theta}_{\text{ft},i}\}_{i=1}^T, \eta$
Output: $\boldsymbol{\theta}_{\text{merged}}$
for $i = 1$ **to** T **do**
   ▷ Get task vector
   $\boldsymbol{\delta}_i \leftarrow \boldsymbol{\theta}_{\text{ft},i} - \boldsymbol{\theta}_{\text{pre}}$
   for $l = 1$ **to** L **do**
      ▷ SVD
      $\boldsymbol{u}_k, \sigma_k, \boldsymbol{v}_k \leftarrow \textbf{SVD}(\boldsymbol{\delta}_i^l)$
      $r \leftarrow \textbf{rank\_keep}(\boldsymbol{\sigma}, \eta, p)$
      ▷ Rescale Singular Values
      for $k = 1$ **to** r **do**
         $\sigma_k' \leftarrow \frac{\|\boldsymbol{\sigma}\|_1}{\|\boldsymbol{\sigma}_{1:r}\|_1} \cdot \sigma_k$
      ▷ Reconstruct
      $\boldsymbol{\delta}_{i,\text{out}} \leftarrow \sum_{k=1}^r \boldsymbol{u}_k \sigma_k' \boldsymbol{v}_k$
▷ Simple Averaging
$\boldsymbol{\delta}_{\text{merged}} \leftarrow \frac{1}{T} \sum_{i=1}^T \boldsymbol{\delta}_{i,\text{out}}$
return $\boldsymbol{\theta}_{\text{merged}} \leftarrow \boldsymbol{\theta}_{\text{pre}} + \boldsymbol{\delta}_{\text{merged}}$

A.3 Discussion on EMR-Merging

EMR-Merging (Huang et al., 2024) is a recent data-free model merging method that reports outstanding performance with minimal additional storage. It first constructs a unified merged task vector, τ_{uni}, which retains the maximum amplitude and sign information shared by all task vectors (τ_i). Then, task-specific masks (M_i) and rescalers (λ_i) are derived based on sign agreement and parameter magnitude alignment between τ_i and τ_{uni}. Finally, during inference, EMR-Merging dynamically adapts τ_{uni} for each task using

$$\hat{W}_t = W_{\text{pre}} + \hat{\tau}_t,$$

where

$$\hat{\tau}_t = \lambda_t \cdot M_t \odot \tau_{\text{uni}}.$$

In other words, EMR-Merging adjusts model weights at run-time, whereas our approach, along with the included baselines (i.e., TIES, MetaGPT, and TALL-masks), operates statically. This makes direct comparison infeasible; therefore, we do not include EMR-Merging as one of the baselines.

A.4 Discussion on DARE

STAR follows a similar protocol to DARE (Yu et al., 2024), as both methods involve two steps: dropping certain components and rescaling. However, there are key differences between them.

On one hand, DARE randomly drops entries of task vectors in parameter space, following:

$$\mathbf{m}^t \sim \text{Bernoulli}(p),$$

$$\tilde{\delta}^t = (1 - \mathbf{m}^t) \odot \delta^t.$$

In contrast, STAR selectively removes redundant dimensions in spectral space.

On the other hand, DARE's rescaling scheme is based on:

$$\hat{\delta}^t = \frac{\tilde{\delta}^t}{1-p},$$

aiming at approximating the original embeddings, while STAR's rescaling focus on restore the spectral-truncated weight matrices to their original scale.

Unlike STAR, which can function as a standalone model merging method, DARE primarily serves as a plug-in to enhance other merging techniques. For comparison, we follow DARE's protocol and report the results of DARE+TA (Task Arithmetic) and DARE+TIES in Table 2. Specifically, we vary DARE's drop rate p from $\{0.1, 0.2, \ldots, 0.9\}$, and the results suggest that even when DARE is applied on top of TA and TIES, STAR still achieves superior performance.

| Method	Hyperparameter	Avg. Normalized
TA	$\alpha = 0.125$	91.67
TA+**DARE**	$\alpha = 0.125, p^* = 0.7$	91.78
TIES	$k = 20$	93.83
TIES+**DARE**	$k = 20, p^* = 0.2$	93.71
STAR	$\eta = 40$	**95.30**

Table 2: Results from merging eight fine-tuned Flan-T5-large models. TA is fixed with a scaling factor of $\alpha = 0.125$, and TIES is set with $k = 20$, using the best-performing DARE drop rate (p^*).

A.5 One-shot STAR performs even better than grid-search TIES

Recall that in Fig. 4, we have shown the one-shot performance with pre-determined $K = 20$ and $\eta = 40$ for TIES and STAR, respectively. In Fig. 7, we further show their best possible results over the grids we searched for. Specifically, from Fig. 7, we see that the grid search does not improve the performance much on Flan-T5-base for both TIES and STAR. Even after performing grid search for TIES, it still fails to surpass the one-shot performance of STAR, further emphasizing the practicality of our
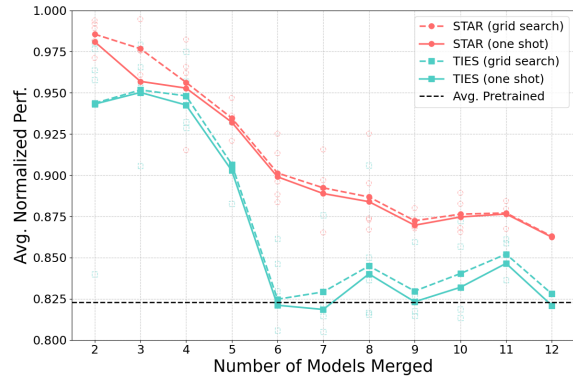
(a) Flan-T5-base
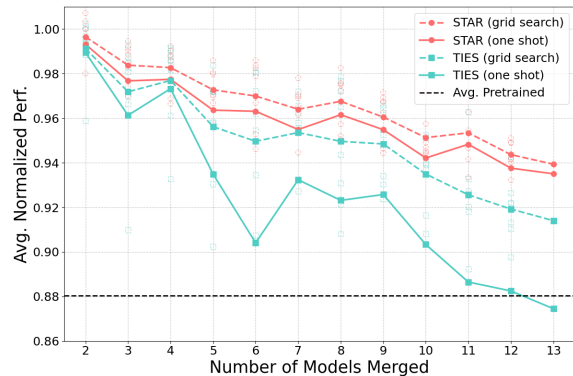
(b) Flan-T5-large

Figure 7: The model merging results on Flan-T5-base and Flan-T5-large with both pre-determined hyperparameter (one-shot, solid lines) and grid-searched hyperparameter (dashed Lines). The performance of each sampled combinations is represented by shaded dots.

method in real-world applications. On Flan-T5-large, the gain from grid search on TIES becomes obvious especially when we are merging more models. With STAR, grid search over η also helps but the results are relatively consistent.

A.6 Details about the fine-tuned models considered in the experiments

For Flan-T5-base, we selected 7 LoRA-16 fine-tuned models from FusionBench[1] (Tang et al., 2024), which is a benchmark targeted for model merging (excluding only CoLA as it tends to output the same answer), and finetuned 5 additional models ourselves on the Finance, IMDB, AG News, HellaSwag, and BoolQ datasets. We applied the same rank (16) and scaling factor (32) as in FusionBench, with the learning rate and number of epochs tuned on the validation set. Following a similar approach, we selected 7 Flan-T5-large models from FusionBench and finetuned 6 additional

---

[1] https://huggingface.co/collections/tanganke

models ourselves, including Finance, IMDB, AG News, HellaSwag, and BoolQ, and PIQA.

For Mistral-Instruct, 20 models are selected from the Lots of LoRA collection [2] (Brüel-Gabrielsson et al., 2024), which encompasses up to 500 diverse task types, making it an ideal environment for evaluating model merging methods. The considered task IDs are: 039, 190, 247, 280, 290, 298, 330, 357, 363, 391, 513, 564, 587, 834, 846, 1198, 1341, 1391, 1448, 1605.

---

[2] https://huggingface.co/Lots-of-LoRAs