A. Feasibility and Convergence Analysis

Proposition 1. *The merging function maintains feasibility, i.e., the merged model stays in the convex hull $\mathcal{M}$.*

Proof. We can rewrite the merging function as:

$$\theta_{t+1} = \big(1 - \sum_{j=1}^{k} \lambda_j^*\big) \cdot \theta_t + \sum_{j=1}^{k} \lambda_j^* \cdot \tilde{s}_j.$$

Since $\theta_t \in \mathcal{M}$ and $\tilde{s}_j \in \mathcal{M}$ for all $j = 1, \dots, k$, and $\{\lambda_j^*\}_{j=1}^k$ are obtained through projection onto the simplex (i.e., $\sum_{j=1}^k \lambda_j^* = 1$ and $\lambda_j^* \geq 0$), we have $\theta_{t+1} \in \mathcal{M}$. This follows from the convexity of $\mathcal{M}$: a convex combination of points in a convex set remains in the set. $\square$

Theorem 1 (Convergence Rate of Soft FW)**.** Consider $\ell(\theta)$ be L-smooth over $\mathcal{M}$, which has two constants: $\mathrm{diam} := \max_{\theta_1,\theta_2\in\mathcal{M}} \|\theta_1 - \theta_2\|$ be the diameter of $\mathcal{M}$, and $\mathrm{subopt} := \ell(\theta_0) - \min_{\theta\in\mathcal{M}} \ell(\theta)$ be the global suboptimality. Consider the soft FW algorithm which introduces the following changes to Algorithm **??**:
1. $\{\tilde{s}_j\}_{j=1}^k$ is the top-k vertices of LMO.
2. $\{\lambda_j^*\}_{j=1}^k = \arg\min_{\lambda\in\Delta^k} \ell(\theta_t + \sum_{j=1}^k \lambda_j(\tilde{s}_j - \theta_t))$.
3. $\theta_{t+1} = \theta_t + \sum_{j=1}^k \lambda_j^*(\tilde{s}_j - \theta_t)$.

We have:

$$\min_{t=0,\dots,T} g_t \leq \frac{\mathrm{subopt}}{T} + \frac{L \cdot \mathrm{diam}^2}{2}.$$

Proof. We first define g_t^k as the top-k FW gap of the soft FW algorithm:

$$g_t^k := \max_{\lambda\in\Delta^k} \max_{s_1,\dots,s_k\in\mathcal{M}} \sum_{j=1}^{k} \lambda_j \langle \nabla\ell(\theta_t), \theta_t - s_j\rangle.$$

Comparing to the full FW gap

$$g_t = \max_{s\in\mathcal{M}} \langle \nabla\ell(\theta_t), \theta_t - s\rangle,$$

we have:

$$g_t^k \geq g_t$$

because the top-k FW gap subsumes the original FW gap by setting $\lambda_1 = 1$ and $\lambda_j = 0$ for $j = 2, \dots, k$. Intuitively, selecting multiple descent directions and optimizing their combination always gives at least as much descent as the single best direction. From the Lipschitz continuity of $\ell(\theta)$, we have:

$$\ell(\theta_{t+1}) \leq \ell(\theta_t) + \langle \nabla\ell(\theta_t), \theta_{t+1} - \theta_t\rangle + \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2.$$

Using the update rule $\theta_{t+1} = \theta_t + \sum_{j=1}^k \lambda_j^*(\tilde{s}_j - \theta_t)$, we have:

$$\langle \nabla\ell(\theta_t), \theta_{t+1} - \theta_t\rangle = -g_t^k.$$

Therefore,

$$\ell(\theta_{t+1}) \leq \ell(\theta_t) - g_t^k + \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2.$$

Since θ_{t+1} is a convex combination of θ_t and $\tilde{s}_j$, we have:

$$\|\theta_{t+1} - \theta_t\|^2 \leq \mathrm{diam}^2.$$

Hence,

$$\ell(\theta_{t+1}) \leq \ell(\theta_t) - g_t^k + \frac{L}{2}\mathrm{diam}^2.$$

Summing over $t = 0, \dots, T-1$, we have:

$$\sum_{t=0}^{T-1} g_t^k \leq \ell(\theta_0) - \ell(\theta_T) + \frac{LT}{2}\mathrm{diam}^2.$$
$$\leq \mathrm{subopt} + \frac{LT}{2}\mathrm{diam}^2.$$

Therefore,

$$\min_{t=0,\dots,T} g_t^k \leq \frac{1}{T}\sum_{t=0}^{T-1} g_t^k \leq \frac{\mathrm{subopt}}{T} + \frac{L}{2}\mathrm{diam}^2.$$

The same result holds for g_t by the definition of g_t^k. $\square$

This convergence proof for non-convex objective functions is based on the proof given by [16]. Due to the soft LMO, we obtain a better convergence rate $O(\frac{1}{T})$ over the vanilla rate $O(\frac{1}{\sqrt{T}})$ with a price to solve a relatively more expensive iteration to obtain the optimal coefficients. This might result in a longer total time, but it is worthy of a solution to the problem of model merging.

B. Data Efficiency

As illustrated in Figure 1, `FW-Merging` outperforms all other model merging methods in terms of performance for the language benchmark. Its performance also surpasses that of traditional MTL while using less training data.

C. Experiment Details

C.1. Benchmarks

Discriminative Tasks. Following previous research [17], 10% of the training split is used as validation split, while the original validation set is used as test set. We fine-tuned 8 RoBERTa on 8 tasks form the GLUE benchmark [31]: QNLI, COLA, STS-B, QQP, SST-2, MRPC, MNLI, RTE. For the evaluation benchmark, we use MNLI, QNLI, QQP, and RTE.

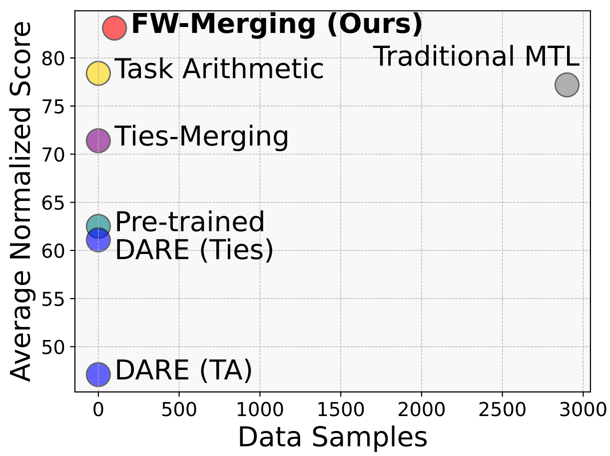

Figure 1. Performance vs. #Data Samples.

Generative Tasks. We collected the following fine-tuned LLaMA2-7B checkpoints from Hugging Face:

- Code Generation[1]
- Medical QA[2]
- News Summarization[3]
- Commonsense Reasoning[4]
- Machine Translation[5]
- Natural Language Understanding[6]

For evaluation, we used the first 1,000 samples from CNN/DM summarization [20], the full test set of PubMedQA [13], and HumanEval [3]. Performance was measured using ROUGE scores for summarization, accuracy for medical QA, and pass@1 accuracy for code generation.

[1] https://huggingface.co/arnavgrg/codealpaca-qlora

[2] https://huggingface.co/SanjanaR01/medical-dialogue-summary-llama2-7b-peft-qlora

[3] https://huggingface.co/ernlavr/llama2_7bn-xsum-cnn-lora-adapter

[4] https://huggingface.co/Styxxxx/llama2_7b_lora-piqa

[5] https://huggingface.co/Styxxxx/llama2_7b_lora-wmt16_translate_roen, https://huggingface.co/Styxxxx/llama2_7b_lora-wmt16_translate_csen, https://huggingface.co/Styxxxx/llama2_7b_lora-wmt16_translate_deen, https://huggingface.co/Styxxxx/llama2_7b_lora-wmt16_translate_fien, https://huggingface.co/Styxxxx/llama2_7b_lora-wmt16_translate_ruen, https://huggingface.co/Styxxxx/llama2_7b_lora-wmt16_translate_tren

[6] https://huggingface.co/Styxxxx/llama2_7b_lora-wnli, https://huggingface.co/Styxxxx/llama2_7b_lora-sst2, https://huggingface.co/Styxxxx/llama2_7b_lora-snli, https://huggingface.co/Styxxxx/llama2_7b_lora-rte, https://huggingface.co/Styxxxx/llama2_7b_lora-qnli, https://huggingface.co/Styxxxx/llama2_7b_lora-cola

Vision Tasks. We use models fine-tuned on the same 20 tasks as [32]: KMNIST [6], EMNIST [8], SVHN [21], GTSRB [26], FER2013 [10], DTD [5], EuroSAT [11], MNIST [9], RenderedSST2 [24, 25], Cars [14], PCAM [30], RESISC45 [4], FashionMNIST [33], SUN397 [34], CIFAR100 [15], Flowers102 [22], Food101 [1], OxfordIIITPet [23], CIFAR10 [15], STL10 [7].

C.2. Baselines

- **Pre-trained**: Employs a pre-trained model for each task without adapting it to the downstream tasks.
- **Individual**: Fine-tunes distinct models for each task, providing the performance upperbound for task-specific performance.
- **Traditional MTL**: Fine-tunes a single model on all tasks, providing a baseline for multi-task learning.
- **Weight Averaging [12]**: Averages the weights of separately fine-tuned models for different tasks, serving as a simple baseline.
- **Task Arithmetic [19]**: Creates a multi-task vector by adding individual task vectors, which are scaled by a coefficient (λ) and incorporated into the pre-trained model's parameters.
- **Fisher Merging [18]**: Uses the Fisher information matrix to determine the importance of model parameters, preserving crucial parameters for each task.
- **Ties-Merging [35]**: Merges models by applying techniques like pruning, parameter sign determination, and separate merging to generate a merged task vector (τ), which is added to the original model's parameters with a scaling factor (λ) tuned on a validation set.
- **AdaMerging [36]**: Adapts merging coefficients at either the task or layer level by minimizing entropy over unlabeled test data, using this as a surrogate objective for model merging.
- **Concrete Merging [28]**: Utilizes a meta-learning framework to generate a concrete mask that mitigates task interference during the merging process.
- **Representation Surgery [37]**: Aligns the representation of the merged model with those of the individual models while adjusting biases to ensure compatibility across tasks.

We used Fusion Bench [29] for evaluation of the vision tasks. We follow the experiment setup provided there. AdaMerging is run with the same setup as detailed in their paper, with a learning rate of 0.001, momentum values of (0.9, 0.999), a batch size of 16, and 500 iterations. Surgery is applied to the merged model from AdaMerging.

C.3. Implementations

On language benchmarks, with the initial solution being the merged model from task arithmetic, and $\texttt{FW}_{\texttt{hard}}$ is run for

10 iterations. On vision tasks, the initial solution is the merged model from AdaMerging, and $\texttt{FW}_{\texttt{hard}}$ runs for 3 iterations. For vision benchmarks, $\texttt{FW}_{\texttt{soft}}$ is run for 15 iterations with the pre-trained model as the initial solution.

For the discriminative language benchmark, 100 data samples from each of MNLI, QNLI, QQP, and RTE are randomly selected as calibration datasets. For generative language tasks, 100 samples are randomly drawn from the training splits of CNN/DM [20], CodeAlpaca-20k [2], and PubMedQA [13]. For vision tasks, training samples are randomly drawn from the training splits of SUN397 [27], Stanford Cars [14], GTSRB [26], and DTD [5].

C.4. Scaling Experiment Setups

For scaling experiments with irrelevant models, we evaluate performance on SUN397 [27], Stanford Cars [14], GTSRB [26], and DTD [5]. The irrelevant models consist of the vision models listed in Appendix C.1, excluding those fine-tuned on these four tasks. For scaling experiments with relevant models, we use all 20 vision tasks as evaluation benchmarks, progressively adding the corresponding fine-tuned models to the merging pool. We employ $\texttt{FW}_{\texttt{soft}}$ for these scaling experiments. To ensure a fair comparison, `FW-Merging` optimizes the merging coefficients using entropy loss on test samples, similar to Adamerging. Adamerging is run for 300 iterations in experiments with irrelevant models and 200 iterations in those with relevant models.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014. 2

[2] Sahil Chaudhary. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca, 2023. 3

[3] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. 2021. 2

[4] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 2

[5] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 2, 3

[6] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018. 2

[7] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 2

[8] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017. 2

[9] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012. 2

[10] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *Neural information processing: 20th international conference, ICONIP 2013, daegu, korea, november 3-7, 2013. Proceedings, Part III 20*, pages 117–124. Springer, 2013. 2

[11] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 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(7):2217–2226, 2019. 2

[12] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 2

[13] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, 2019. 2, 3

[14] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 2, 3

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[16] Simon Lacoste-Julien. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016. 1

[17] Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration

of modular expertise in model merging. *Advances in Neural Information Processing Systems*, 37:78905–78935, 2025. 1

[18] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022. 2

[19] Eric Mitchell, Kenton Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022. 2

[20] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016. 2, 3

[21] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, page 4. Granada, 2011. 2

[22] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 2

[23] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 2

[24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[25] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013. 2

[26] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011. 2, 3

[27] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems*, 33:8728–8740, 2020. 3

[28] Anke Tang, Li Shen, Yong Luo, Liang Ding, Han Hu, Bo Du, and Dacheng Tao. Concrete subspace learning based interference elimination for multi-task model fusion. *arXiv preprint arXiv:2312.06173*, 2023. 2

[29] Anke Tang, Li Shen, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. FusionBench: A Comprehensive Benchmark of Deep Model Fusion, 2024. 2

[30] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *Medical image computing and computer assisted intervention–mICCAI 2018: 21st international conference, granada, Spain, September 16-20, 2018, proceedings, part II 11*, pages 210–218. Springer, 2018. 2

[31] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018. 1

[32] Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. Localizing task information for improved model merging and compression. *arXiv preprint arXiv:2405.07813*, 2024. 2

[33] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 2

[34] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119:3–22, 2016. 2

[35] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023. 2

[36] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. *arXiv preprint arXiv:2310.02575*, 2023. 2

[37] Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. Representation surgery for multi-task model merging. *arXiv preprint arXiv:2402.02705*, 2024. 2