LLM Merging: Building LLMs Efficiently through Merging

Derek Tam* Margaret Li* Prateek Yadav* Rickard Brüel-Gabrielsson* Jiacheng Zhu*
Kristjan Greenewald Mikhail Yurochkin Mohit Bansal Colin Raffel
Leshem Choshen
llm-merging@googlegroups.com

Abstract

Training high-performing large language models (LLMs) from scratch is a notoriously expensive and difficult task, costing hundreds of millions of dollars in compute alone [1, 72, 70]. These pretrained LLMs, however, can cheaply and easily be adapted to new tasks via fine-tuning, leading to a proliferation of models that suit specific use cases. Recent work has shown that specialized fine-tuned models can be rapidly *merged* to combine capabilities and generalize to new skills. This raises the question: given a new suite of desired skills and design parameters, is it necessary to fine-tune or train yet another LLM from scratch, or can similar existing models be re-purposed for a new task with the right selection or merging procedure? The LLM Merging challenge aims to spur the development and evaluation of methods for merging and reusing existing models to form stronger new models without needing additional training. Specifically, the competition focuses on merging existing publicly-released expert models from Hugging Face, using only minimal compute and additional parameters. The goal will be to develop merged models that outperform existing models and existing merging baselines. Submissions will be judged based on the average accuracy on a set of held-out multiple-choice evaluation tasks and their efficiency. To make the competition as accessible as possible and ensure that the merging procedures are more efficient than fine-tuning, we will enforce a compute budget and focus on merging models with fewer than 8B parameters. A starter kit with all necessary materials (baseline implementations, requirements, the evaluation script, etc.) will be released on May 1st.

https://llm-merging.github.io/

Keywords

Model Recycling, Merging, Mixture of Experts, Routing, Model Selection

1 Competition description

1.1 Background and impact

This competition addresses the question: *"Can we develop better methods for composing and recycling models for generalization?"* Composing models [49, 78, 17] is the problem of efficiently combining or "merging" a set of expert models into one. We specifically focus on the use case where the expert models are trained on various tasks and the merged model is expected to generalize to new tasks. Given a set of base models and their variations fine-tuned for different tasks, merging

*Equal leaders

Preprint. Under review.

combines (a subset of) these models to form a single model that can perform well on new tasks. To be worthwhile, performing the merge should be cheaper than re-training a model from scratch on all the data used to train the original models, and may even avoid any forward or backward passes in the models. By recycling the compute used to train the original models, merging opens a pathway for (very) efficient development of performant models as well as collaborative approaches that continually improve models over time [23].

This competition seeks to advance the development of merging methods. The development of practical and technically-sound merging methods would lower the amount of resources and compute required, and allow for more researchers and practitioners to participate in the development of state-of-the-art models.

The LLM community has developed many methods that can be viewed as forms of model merging:

- **Parameter averaging** such as simple averaging [50, 66], Fisher merging [49] which weights parameters based on their importance, TaskArithmetic [37] which uses linear combinations of task vectors to construct the merged model, and TIES-Merging [83] which accounts for parameter interference when averaging. Variations of parameter averaging merging methods are also often applied to merge Stable DiffusionXL [57] models individually fine-tuned on different styles to generate images combining these styles [61].
- **MoE-based merging,** e.g., merging of experts in an MoE model to speed up inference [46], merging models into a new MoE model with token-based gating [31], and learning a post-hoc router among specialized PEFT modules [53].
- **Model stacking** methods apply LLMs sequentially or combine their outputs. Among them, self-correctors [77] and aligners [54, 39] pass generations of one LLM as an input to another to produce the final output; proxy-tuning [47] combines next-token predictions from a larger base and smaller fine-tuned LLMs; and controlled decoding [52] adjusts next token probabilities to align them with a reward function via a prefix scoring LLM.
- **Model zipping** combines layers of different LLMs, e.g., CALM [3] via cross-attention and Frankenllama [30] by extending the hidden dimensions.
- **Model routing** methods aim to select the best LLM for each input via a smaller routing model while keeping the candidate LLMs intact. For example, LLM Blender [41] trains a small model to rank candidate generations, FoE [74] uses token embeddings from candidate expert LLMs to select the best one, and [63, 35] train meta-models to predict if an LLM will produce good generation for an input [48].

Many of these recent works have been in the NeurIPS community [83, 42, 64, 46, to name a few], demonstrating the relevance of this problem. Besides the NeurIPS community, we expect the competition's aim to efficiently develop performant models will be of great interest to a broad audience. This can be seen on the Hugging Face leaderboard [4] where many of the top-performing models are merges of other models – in fact, merging is so popular that a filter for merging had to be created to declutter the leaderboard [2]. Finally, we design the competition to be open to as broad of an audience as possible by purposefully using an extremely large list of models and place no constraints on how models can be merged except that the number of parameters has to be less than $8B$. The models will be evaluated based on the average accuracy on some held out tasks and their efficiency.

1.2 Novelty

This merging competition is, to the best of our knowledge, the first competition focused on the fast-growing field of model merging. The most closely related competitions are on efficient or cognitively plausible pretraining (the BabyLM Competition [76]) and on efficiently finetuning models [60], both of which share a focus on efficiency with our competition. However, such competitions focus on *training* procedures, whereas our focus is on reusing current models with no or minimal training. There has been significant work on better training procedures - usually done via selecting better dataset mixtures, better training objectives, or faster training via lower-precision training or higher quality data [28]. Merging, on the other hand, has been much less explored so far. Current works focus on better merging objectives, similarities, and interference between different models or

[2] https://x.com/clefourrier/status/1737184323764167162?s=20

leveraging linear mode connectivity in the loss landscape [80, 83, 25, 24, 49, 51, 46, 78, 17, 2, 69, 42, 37, 45, 43, 5, 29, 38, 67, 84, 68, 59]. Hence our competition is also likely to attract a different community of participants than the prior competitions focusing on efficient training.

1.3 Data

Expert Model: The competition will provide the participants with a list of "expert" models that have already been trained on a task-specific dataset. All of these models will be publicly available on the Hugging Face Model Hub with licenses that permit their use for research purposes. These models can either be fully fine-tuned models or models obtained by parameter-efficient fine-tuning methods such as LoRA [34]. Models on this list will be required to satisfy the following criteria: (1) model size $\leq$ 8B parameters, and (2) model with licenses compatible with research use (e.b., MIT, Apache, etc). The goal of this competition is to re-use the provided models to create a generalist model that can perform well on a wide variety of skills like reasoning, coding, maths, chat, and tool use. This list of models will include popular pre-trained models such as LLaMA-7B [72], Mistral-7B [40], and Gemma-7B [71].

Datasets: Along with these expert models, we also plan to provide a set of *validation tasks* that can be used to evaluate the final method's performance. The datasets will be released as part of the starter kit for the participants and are already hosted on the Hugging Face Hub with a permissive license. These datasets are meant to ensure the merging method runs under the time limit. Apart from these, we will have two sets of hidden tasks that will be used to evaluate the submissions from participants: (1) a set of *leaderboard ranking test tasks*, and (2) a set of *final ranking test tasks*. The leaderboard ranking tasks will have some overlap with the test set tasks to provide an additional signal to the participants.

We will not collect or release any new datasets for training or evaluation as part of this competition.

Validation Datasets List:

1. `MAWPS` [44]: MIT License
2. `BoolQ` [19]: CC BY-SA 3.0

1.4 Tasks and application scenarios

Real-World Problem: This competition focuses on developing techniques for efficiently reusing already available expert models trained by others. This directly addresses a critical challenge faced by both academia and industry.

- **Academia:** Research labs often lack the resources to train massive models from scratch. Merging allows them to leverage existing models, accelerating research and development.
- **Industry:** Companies invest heavily in training large models. Merging offers a way to "recycle" this compute investment by incorporating past models into new ones, promoting resource efficiency and cost savings.

Benefits of Model Merging: There are several benefits of reusing models, some of which are listed below.

1. **Reduced Compute Costs:** Reuse of pre-trained or fine-tuned models leads to significant computational efficiency gains, lower costs, and reduced environmental impact.
2. **Compositional Generalization:** Merged models can potentially learn new tasks by combining knowledge from existing ones [3]. This helps to ameliorate the problem of identifying optimal data mixtures when training a model for multiple skills/datasets.
3. **Modular Design:** Smaller, specialized models can be easier to optimize and adapt for specific tasks resulting in better performance.
4. **Privacy Advantages:** Contributors with private data are often more comfortable sharing a model fine-tuned on their data than they are sharing the data itself.
5. **Decentralized and Continual Learning:** Merging enables collaborative development where individual labs contribute specialized models to improve a general system.

While the problem of merging models – i.e., combining models trained on different tasks in a way that retains the capabilities of the original models – appears challenging, an abundance of previous works has succeeded in steadily increasing the performance of merging methods [17, 49, 37, 83], providing evidence for the promising potential of the approach.

The final objective is to have a system that reuses many existing models – for example, a small base model like LLaMA-7B [72], along with multiple fine-tuned versions of it each of which specializes in a different domain – to perform well on general LLM benchmarks and rival the performance of much larger models like GPT-4 on the most prominent use-cases. Moreover, this system should continually [79, 81] and easily adapt to any new use cases.

1.5 Metrics

Competition rankings will be decided by the final evaluation metric, calculated as average accuracy on a set of held-out evaluation tasks. The performance for each task will be the raw accuracy on the task. There is no applicable notion of normalized accuracy since we do not have an upper-bound performance for each task. All held-out tasks will be weighted equally for simplicity.

All tasks are multiple-choice, so each model's score will be deterministic. No random seeds and no error bars are needed. The organizers plan to reproduce submissions to ensure that each merging method works as expected and fits under the compute constraint. We will be releasing all evaluation code, as detailed in the next section. Additionally, we will include a requirements file listing all pre-installed packages and relevant software and hardware used for evaluation. Participants will be required to submit their own requirements file for any additional necessary packages. Evaluation will be run on the hardware and software listed, by directly calling only the scripts we will release to participants.

1.6 Baselines, code, and material provided

The baseline includes parameter averaging with different base models. Baseline results for the held-out tasks will not be posted at the beginning of the competition, to avoid leakage of information about the held-out tasks, however, baseline results on all development tasks will be released, and baseline results on test tasks will be released at the conclusion of the competition, along will all other results, when the test tasks themselves are revealed.

A "starting kit" will be released on May 1st. This kit will include the baseline implementations listed above, as well as a package requirements file, code stubs, the evaluation script, and lists of models and development tasks as described in §1.4. With this kit, participants will be able to use a single command to install requirements, and to launch evaluation of baseline methods on the development tasks, as well as any other tasks in the Hugging Face Datasets Hub. By filling in the code stubs with their merging methods, participants can easily evaluate during development.

1.7 Website, tutorial and documentation

Website: The competition website, https://llm-merging.github.io/, will serve as the central hub for all information, including the competition timeline, registration process, step-by-step instructions, and the protocol for submission and evaluation. The website is already live, and all the final details will be available within two weeks of acceptance notification, ensuring participants have enough time to review the materials and prepare for the competition.

Tutorial and documentation: Apart from the main competition website, we also host a documentation page for **FAQ/Tutorial** at https://llm-merging.readthedocs.io/. The documentation page is supported by the widely used open-source documentation platform **Read the Docs** so that participants can quickly get started with environment setup, code implementation, and model submission. Moreover, we provided a starter kit with an end-to-end submission flow at https://github.com/llm-merging/LLM-Merging.

Communication: We provide the following google group: llm-merging@googlegroups.com for all participants to receive updates and reminders. We also host a competition Discord server (https://discord.gg/ufycruJx) to allow discussions, asking questions, finding teammates and providing feedback.

2 Organizational aspects

2.1 Protocol

In this competition, participants will utilize model and parameter-efficient adapters, provided through Hugging Face, to address specific tasks. These tasks are represented in a dataset provided to all participants, reflecting similar challenges in a withheld test dataset. The main restriction is the computational budget of 1 hour on a single A6000 GPU. Therefore, we suspect that the best performing submission will combine the available models and adapters into a more high performing merged model with less compute than fully fine-tuning any large model.

Participation Steps:

1. Registration: All participants must register on our official competition website. At the start of the competition, the starter code will be available on the competition's GitHub page.
2. Environment Setup: Participants should ensure their own development environment meets the requirements listed on our competition page.
3. Dataset Download: Once registered and the competition begins, participants can download the initial dataset from the competition website. This dataset contains examples from validation and re-calibration tasks, corresponding to tasks in the withheld test data.
4. Development: Utilize the provided adapters and dataset to develop your solution. Participants' scripts must adhere to the specified computational budget constraints.
5. Submission: Participant submits their script via the submission form on our competition website. Please include any necessary documentation and ensure your script is executable within the defined computational environment. If it is not executable, an error will be returned within 5 minutes of submission.

Submission Requirements:

1. A single executable script.
2. Documentation explaining the approach and any assumptions made.
3. A requirements.txt file listing any external libraries used (beyond the ones provided in the starter code).

Evaluation Criteria:

Submissions will be automatically run within a controlled environment against the withheld test dataset. They will be evaluated based on:

1. Accuracy of the results against the test dataset.
2. Efficiency and resource usage within the given computational budget.

Competition Phases:

1. Development Phase: Participants develop their solutions using the provided dataset.
2. Submission Phase: Participants submit their scripts and documentation.
3. Evaluation Phase: Submissions are evaluated against the test dataset.
4. Review Phase: Top submissions undergo a manual review to ensure compliance with competition rules.

Leaderboard and Results:

A leaderboard will be maintained on the competition website, updated regularly with participants' scores. Final results will be announced at the conclusion of the Evaluation Phase.

Cheating and Overfitting Prevention:

1. Regular audits of submission logs to detect abnormal activity. E.g. downloading not-allowed pre-trained models.

2. Use of a withheld test dataset to prevent overfitting.
3. Manual review of top-scoring submissions to verify integrity and adherence to competition rules.

Beta Tests and Dry Runs:

Prior to the official start of the competition, we will conduct a series of beta tests and dry runs:

1. Beta Test: Selected participants (mainly the organizers) will run through the competition process to identify and resolve any issues.
2. Dry Run: A mock competition using a separate dataset to ensure the submission and evaluation processes function as intended. Participants and interested parties are encouraged to report any issues or concerns during these preliminary phases.

2.2 Rules and Engagement

Rules given to the competitors The competition calls for building the best model on a diverse set of tasks. The winning model will be the one that has the best results on the hidden test tasks. This test would call for general abilities such as reasoning, text generation, question answering, etc.

We encourage each team to share a report on any approaches or insights they found. The report can range from a technical description to more analysis or evaluation of various methods. We ask each team that submits a report to additionally give feedback on the reports from 2 other teams to ensure the interesting findings of the competition are shared and documented.

1. Participants may only merge, combine, and mix models on the list provided. No other models may be used.
2. We encourage model selection efforts, but one may not submit one of the models in the list.
3. There are no restrictions on what can be done when merging models. One can keep, add or delete layers to merge, mix and match different architectures.
4. The goal of the competition is making the best use we know from multiple models, not data synthesis or fine-tuning of individual models, hence computation will be limited to an hour on a single GPU for training and evaluation together.
5. Evaluation will be performed on a few thousand examples.
6. For submissions that break some of the above rules but follow the spirit of the competition, we encourage participants to notify the organizers so that we can recognize such submissions and report the findings without allowing them to compete for the monetary prizes.
7. Evaluation, which includes the full process of merging and evaluating, will be time-limited.
8. For each team, we cap the number of submissions (excluding the final submission) to the leaderboard for evaluation to 5.

How the rules lead to the right outcomes The goal of this competition is to develop better and novel methods for merging and reusing models to enable better generalization. To encourage novelty, this competition has very little constraints to allow for diverse usage of models. While most current works deal with merging models of the same architecture, we have no such constraints.

Additionally, model selection is often overlooked in the literature, with previous works assuming that only a small set of models is available. We remove this constraint as well and consider a much larger pool. Placing little constraints on participants opens different possibilities for participants to explore. Finally, the mandatory report should allow for knowledge learned to be propagated.

We restrict computation to prevent methods that win simply because of extensive training rather then developing better model merging techniques. Since the leaderboard will be continuously updated, to prevent teams from overfitting, we limit the number of submissions per team to 5.

The models selected focus on 3 aspects: diversity, compute budget, and quality. We constrain the models allowed to a pre-specified list to prevent submissions from gaming the competition by uploading or using a model that does not adhere to other rules. As there is normally a trade-off between quality and compute, we restricted the size of the allowed models to 7.9B parameters. This allows enough memory for inference on a single GPU with a few models (e.g., in a small mixture of experts or a new model by combining two architectures), or a vast number of LoRAs [34].

Communicate with the Organizers We will have a dedicated Discord channel in which participants can ask questions and the organizers can announce important updates.

2.3 Schedule and readiness

- Early June - Release models and evaluation script
- Mid-September - submissions due
- Mid Oct - Organizers run evaluation scripts and participatnts submit reports
- Beg Nov - Announcement of Winner
- Mid-December - Competition Presentation

2.4 Competition promotion and incentives

The competition would be promoted in our active accounts on Twitter, Mastodon, Bluesky, Weibo, etc. Some of the organizers have quite active and popular accounts as well as have advertised previous competitions and shared tasks.

We plan to allow participants to write a report and make it public. Moreover, we have sponsorship from Arcee AI, Sakana AI, and HuggingFace of USD 8K, 3K and 2K for the first, second, and third place as well as a 1K award for the most interesting finding awarded by the organizing committee and 1K prize for a GPU budget-friendly submission. The first three prizes will be awarded by the scores on the hidden eval set while the second will be chosen by the organizers. The last prize follows the spirit and motivation of the competition. As the competition aims to develop new methods where anyone can improve models or develop methods that utilize a diversity of expertise, we will encourage participants to make computation and memory-efficient submissions and give the last award to the optimal resource submission that achieves a minimum threshold.

3 Resources

3.1 Organizing team

Derek Tam dtam@cs.toronto.edu
Derek Tam is a PhD student at the University of Toronto advised by Colin Raffel. He has previously worked on model merging [69, 83].

Margaret Li margsli@cs.washington.edu
Margaret Li is a PhD student at University of Washington advised by Luke Zettlemoyer and Tim Althoff, and is concurrently a Visiting Researcher at FAIR (Meta). She has previously worked on efficient training of and merging of independent expert models specialized to text domains [45, 32, 6].

Prateek Yadav praty@cs.unc.edu
Prateek Yadav is a Ph.D. student at the University of North Carolina at Chapel Hill advised by Mohit Bansal. He has previously worked on model merging [83], continual learning [79, 81], mixture-of-expert models [46], and training/inference efficiency [80, 46].

Rickard Brüel-Gabrielsson brg@mit.edu
Rickard Brüel-Gabrielsson is currently pursuing his doctoral studies at MIT within CSAIL. His academic research primarily focuses on LORA [89], unsupervised learning [26, 8], self-supervised learning [9], and graph learning [7, 27]. Additionally, he is the creator of and head instructor for the course on Foundation Models and Generative AI at MIT.

Dr. Jiacheng Zhu zjc@mit.edu
Jiacheng Zhu is a postdoctoral researcher at MIT CSAIL. His research focuses on developing generalizable and trustworthy machine learning models. One line of his research investigates fine-tuning and transfer of models. He has studied the theoretical understanding of LoRA [89] as well as model transfer through the lens of optimal transport theories [87]. Another line of his work exploits data distribution mixtures that promote robustness to distribution shifts [36, 88]. Part of Jiacheng's PhD research at Carnegie Mellon University was sponsored by the Qualcomm Innovation Fellowship in 2022. Jiacheng has co-organized competitions such as the SeasonDepth Challenge, a deep learning perception competition.

Dr. Kristjan Greenewald kristjan.h.greenewald@ibm.com
Kristjan Greenewald is a research scientist and PI at the MIT-IBM Watson AI Lab (IBM Research), with research focusing on applying statistical techniques towards understanding and improving generative modeling. Prior to joining MIT-IBM, he received his PhD from the University of Michigan and completed a postdoctoral fellowship in the Statistics Department at Harvard University. His relevant work includes Bayesian model merging [86, 85], theoretical understanding of LoRA [89], as well as risk-aware evaluation of language models [55] and LLM uncertainty calibration [62].

Dr. Mikhail Yurochkin mikhail.yurochkin@ibm.com
Mikhail Yurochkin is a research scientist and manager at the MIT-IBM Watson AI Lab where he leads the Statistical Large Language Modeling group. Before joining IBM, he completed PhD in Statistics at the University of Michigan, advised by XuanLong Nguyen. His research focuses on the challenges of reliable and efficient use of LLMs. His works include model fusion [85, 86, 73, 18], efficient evaluation of LLMs [58], and various methods for routing [74, 63] and stacking [54] LLMs.

Dr. Mohit Bansal mbansal@cs.unc.edu
Dr. Mohit Bansal is the John R. & Louise S. Parker Professor and the Director of the MURGe-Lab (UNC-NLP Group) in the Computer Science department at UNC Chapel Hill. He received his PhD from UC Berkeley in 2013 and his BTech from IIT Kanpur in 2008. His research expertise is in natural language processing and multimodal machine learning, with a particular focus on multimodal generative models, grounded and embodied semantics, faithful language generation, and interpretable and generalizable deep learning. He is a recipient of IIT Kanpur Young Alumnus Award, DARPA Director's Fellowship, NSF CAREER Award, Google Focused Research Award, Microsoft Investigator Fellowship, Army Young Investigator Award (YIP), DARPA Young Faculty Award (YFA), and outstanding paper awards at ACL, CVPR, EACL, COLING, and CoNLL. He has been a keynote speaker for the AACL 2023, CoNLL 2023, and INLG 2022 conferences. His service includes ACL Executive Committee, ACM Doctoral Dissertation Award Committee, CoNLL Program Co-Chair, ACL Americas Sponsorship Co-Chair, and Associate/Action Editor for TACL, CL, IEEE/ACM TASLP, and CSL journals. He is also serving as the program chair for EMNLP 2024.

Dr. Colin Raffel colin.raffel@vectorinstitute.ai
Colin Raffel is an Associate Professor at the University of Toronto and Associate Research Director at the Vector Institute, specializing in machine learning. His research delves into enabling decentralized collaborative development of models through modular architectures, efficient communication methods for updates, and merging techniques. Additionally, he works on refining training recipes for greater efficiency while also identifying and mitigating risks associated with the deployment of large-scale models, thus contributing significantly to the advancement of machine learning applications across various domains.

Dr. Leshem Choshen leshem.choshen@mail.huji.ac.il
Leshem Choshen is a postdoctoral researcher at MIT/IBM, aiming to collaboratively pretrain through model recycling [22, 82], efficient evaluation [16, 56], and manageable pretraining research (e.g., co-organizing the babyLM shared task [75]). Leshem performed as the publicity chair of EACL2022 and CoNLL2021, organized previous shared tasks and ISCOL 2017. Before leading a small research group at IBM, he received the postdoctoral Rothschild and Fulbright fellowships as well as IAAI and Blavatnik best Ph.D. awards. With broad NLP and ML interests, he also worked on Reinforcement Learning, and Understanding of how neural networks learn [15, 20], with a specific interest in evaluation [13, 14], evaluation of evaluation [11, 10], reference-less metrics [12, 33], quality estimation [21] and related topics. In parallel, he participated in Project Debater, creating a machine that could hold a formal debate, ending in a Nature cover and live debate [65].

3.2 Resources provided by organizers

We are applying for cloud credits to run the evaluation on our side (to preserve the secret evaluation). We plan to use Google Colab to evaluate the merging scripts to allow for an easy replicable platform for both participants and organizers. We have funding secured from a mix of the organizer's institutional affiliations (supporting winner and diversity prize).

3.3 Support requested

We request a place to share the reports by the competitors.

References

[1] 2024. URL https://en.wikipedia.org/wiki/GPT-4.

[2] S. K. Ainsworth, J. Hayase, and S. Srinivasa. Git re-basin: Merging models modulo permutation symmetries. arXiv preprint arXiv:2209.04836, 2022.

[3] R. Bansal, B. Samanta, S. Dalmia, N. Gupta, S. Vashishth, S. Ganapathy, A. Bapna, P. Jain, and P. Talukdar. Llm augmented llms: Expanding capabilities through composition. arXiv preprint arXiv:2401.02412, 2024.

[4] E. Beeching, C. Fourrier, N. Habib, S. Han, N. Lambert, N. Rajani, O. Sanseviero, L. Tunstall, and T. Wolf. Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard, 2023.

[5] G. Benton, W. Maddox, S. Lotfi, and A. G. G. Wilson. Loss surface simplexes for mode connecting volumes and fast ensembling. In International Conference on Machine Learning, pages 769–779. PMLR, 2021.

[6] T. Blevins, T. Limisiewicz, S. Gururangan, M. Li, H. Gonen, N. A. Smith, and L. Zettlemoyer. Breaking the curse of multilinguality with cross-lingual expert language models, 2024.

[7] R. Brüel Gabrielsson. Universal function approximation on graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 19762–19772. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e4acb4c86de9d2d9a41364f93951028d-Paper.pdf.

[8] R. Brüel Gabrielsson and G. Carlsson. Exposition and interpretation of the topology of neural networks. In 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), pages 1069–1076, 2019. doi: 10.1109/ICMLA.2019.00180.

[9] R. Brüel-Gabrielsson, T. Wang, M. Baradad, and J. Solomon. Deep augmentation: Self-supervised learning with transformations in activation space, 2024.

[10] L. Choshen and O. Abend. Automatic metric validation for grammatical error correction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1372–1382, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1127. URL https://aclanthology.org/P18-1127.

[11] L. Choshen and O. Abend. Inherent biases in reference-based evaluation for grammatical error correction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 632–642, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1059. URL https://aclanthology.org/P18-1059.

[12] L. Choshen and O. Abend. Reference-less measure of faithfulness for grammatical error correction. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 124–129, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2020. URL https://aclanthology.org/N18-2020.

[13] L. Choshen and O. Abend. Automatically extracting challenge sets for non-local phenomena in neural machine translation. In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pages 291–303, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1028. URL https://aclanthology.org/K19-1028.

[14] L. Choshen, D. Nikolaev, Y. Berzak, and O. Abend. Classifying syntactic errors in learner language. In Proceedings of the 24th Conference on Computational Natural Language Learning, pages 97–107, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.conll-1.7. URL https://aclanthology.org/2020.conll-1.7.

[15] L. Choshen, G. Hacohen, D. Weinshall, and O. Abend. The grammar-learning trajectories of neural language models. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8281–8297, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.568. URL https://aclanthology.org/2022.acl-long.568.

[16] L. Choshen, E. Venezian, S. Don-Yehia, N. Slonim, and Y. Katz. Where to start? analyzing the potential value of intermediate models. arXiv preprint arXiv:2211.00107, 2022.

[17] L. Choshen, E. Venezian, N. Slonim, and Y. Katz. Fusing finetuned models for better pretraining. arXiv preprint arXiv:2204.03044, 2022.

[18] S. Claici, M. Yurochkin, S. Ghosh, and J. Solomon. Model fusion with kullback-leibler divergence. In International conference on machine learning, pages 2038–2047. PMLR, 2020.

[19] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

[20] A. Y. Din, T. Karidi, L. Choshen, and M. Geva. Jump to conclusions: Short-cutting transformers with linear transformations. ArXiv, abs/2303.09435, 2023. URL https://api.semanticscholar.org/CorpusID:257557722.

[21] S. Don-Yehiya, L. Choshen, and O. Abend. PreQuEL: Quality estimation of machine translation outputs in advance. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11170–11183, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.767. URL https://aclanthology.org/2022.emnlp-main.767.

[22] S. Don-Yehiya, E. Venezian, C. Raffel, N. Slonim, Y. Katz, and L. Choshen. Cold fusion: Collaborative descent for distributed multitask finetuning. ArXiv, abs/2212.01378, 2022. URL https://api.semanticscholar.org/CorpusID:254220858.

[23] S. Don-Yehiya, E. Venezian, C. Raffel, N. Slonim, and L. Choshen. ColD fusion: Collaborative descent for distributed multitask finetuning. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 788–806, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.46. URL https://aclanthology.org/2023.acl-long.46.

[24] R. Entezari, H. Sedghi, O. Saukh, and B. Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. ArXiv, abs/2110.06296, 2021.

[25] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin. Linear mode connectivity and the lottery ticket hypothesis. ArXiv, abs/1912.05671, 2020.

[26] R. B. Gabrielsson, B. J. Nelson, A. Dwaraknath, and P. Skraba. A topology layer for machine learning. In S. Chiappa and R. Calandra, editors, Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, volume 108 of Proceedings of Machine Learning Research, pages 1553–1563. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/gabrielsson20a.html.

[27] R. B. Gabrielsson, M. Yurochkin, and J. Solomon. Rewiring with positional encodings for graph neural networks. Transactions on Machine Learning Research, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=dn3ZkqG2YV.

[28] S. Y. Gadre, G. Ilharco, A. Fang, J. Hayase, G. Smyrnis, T. Nguyen, R. Marten, M. Wortsman, D. Ghosh, J. Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. Advances in Neural Information Processing Systems, 36, 2024.

[29] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. Advances in neural information processing systems, 31, 2018.

[30] C. Goddard. On frankenllama, 2023. URL https://goddard.blog/posts/frankenllama/.

[31] C. Goddard. Mixture of experts for clowns (at a circus), 2023. URL https://goddard.blog/posts/clown-moe/.

[32] S. Gururangan, M. Li, M. Lewis, W. Shi, T. Althoff, N. A. Smith, and L. Zettlemoyer. Scaling expert language models with unsupervised domain discovery, 2023.

[33] O. Honovich, L. Choshen, R. Aharoni, E. Neeman, I. Szpektor, and O. Abend. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7856–7870, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.619. URL https://aclanthology.org/2021.emnlp-main.619.

[34] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. In International Conference on Learning Representations.

[35] Q. J. Hu, J. Bieker, X. Li, N. Jiang, B. Keigwin, G. Ranganath, K. Keutzer, and S. K. Upadhyay. Routerbench: A benchmark for multi-llm routing system. arXiv preprint arXiv:2403.12031, 2024.

[36] P. Huang, M. Xu, J. Zhu, L. Shi, F. Fang, and D. Zhao. Curriculum reinforcement learning using optimal transport via gradual domain adaptation. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, Advances in Neural Information Processing Systems, 2022. URL https://openreview.net/forum?id=_cFdPHRLuJ.

[37] G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, and A. Farhadi. Editing models with task arithmetic. arXiv preprint arXiv:2212.04089, 2022.

[38] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. arXiv preprint arXiv:1803.05407, 2018.

[39] J. Ji, B. Chen, H. Lou, D. Hong, B. Zhang, X. Pan, J. Dai, and Y. Yang. Aligner: Achieving efficient alignment through weak-to-strong correction. arXiv preprint arXiv:2402.02416, 2024.

[40] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. arXiv preprint arXiv:2310.06825, 2023.

[41] D. Jiang, X. Ren, and B. Y. Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. arXiv preprint arXiv:2306.02561, 2023.

[42] X. Jin, X. Ren, D. Preotiuc-Pietro, and P. Cheng. Dataless knowledge fusion by merging weights of language models. arXiv preprint arXiv:2212.09849, 2022.

[43] K. Jordan, H. Sedghi, O. Saukh, R. Entezari, and B. Neyshabur. Repair: Renormalizing permuted activations for interpolation repair. arXiv preprint arXiv:2211.08403, 2022.

[44] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. arXiv preprint arXiv:1705.03551, 2017.

[45] M. Li, S. Gururangan, T. Dettmers, M. Lewis, T. Althoff, N. A. Smith, and L. Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models. arXiv preprint arXiv:2208.03306, 2022.

[46] P. Li, Z. Zhang, P. Yadav, Y.-L. Sung, Y. Cheng, M. Bansal, and T. Chen. Merge, then compress: Demystify efficient SMoe with hints from its routing policy. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=eFWG9Cy3WK.

[47] A. Liu, X. Han, Y. Wang, Y. Tsvetkov, Y. Choi, and N. A. Smith. Tuning language models by proxy. arXiv preprint arXiv:2401.08565, 2024.

[48] K. Lu, H. Yuan, R. Lin, J. Lin, Z. Yuan, C. Zhou, and J. Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. arXiv preprint arXiv:2311.08692, 2023.

[49] M. Matena and C. Raffel. Merging models with fisher-weighted averaging. arXiv preprint arXiv:2111.09832, 2021.

[50] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, 2017.

[51] S. I. Mirzadeh, M. Farajtabar, D. Gorur, R. Pascanu, and H. Ghasemzadeh. Linear mode connectivity in multitask and continual learning. ArXiv, abs/2010.04495, 2021.

[52] S. Mudgal, J. Lee, H. Ganapathy, Y. Li, T. Wang, Y. Huang, Z. Chen, H.-T. Cheng, M. Collins, T. Strohman, et al. Controlled decoding from language models. arXiv preprint arXiv:2310.17022, 2023.

[53] M. Muqeeth, H. Liu, Y. Liu, and C. Raffel. Learning to route among specialized experts for zero-shot generalization. arXiv preprint arXiv:2402.05859, 2024.

[54] L. Ngweta, M. Agarwal, S. Maity, A. Gittens, Y. Sun, and M. Yurochkin. Aligners: Decoupling llms and alignment. arXiv preprint arXiv:2403.04224, 2024.

[55] A. Nitsure, Y. Mroueh, M. Rigotti, K. Greenewald, B. Belgodere, M. Yurochkin, J. Navratil, I. Melnyk, and J. Ross. Risk assessment and statistical significance in the age of foundation models. arXiv preprint arXiv:2310.07132, 2023.

[56] Y. Perlitz, E. Bandel, A. Gera, O. Arviv, L. Ein-Dor, E. Shnarch, N. Slonim, M. Shmueli-Scheuer, and L. Choshen. Efficient benchmarking (of language models). ArXiv, abs/2308.11696, 2023. URL https://api.semanticscholar.org/CorpusID:261076362.

[57] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. Sdxl: improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952, 2023.

[58] F. M. Polo, L. Weber, L. Choshen, Y. Sun, G. Xu, and M. Yurochkin. tinybenchmarks: evaluating llms with fewer examples. arXiv preprint arXiv:2402.14992, 2024.

[59] A. Ramé, K. Ahuja, J. Zhang, M. Cord, L. Bottou, and D. Lopez-Paz. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In International Conference on Machine Learning, pages 28656–28679. PMLR, 2023.

[60] M. Saroufim, W. Yang, J. Isaacson, L. Antiga, D. Guessous, G. Bowyer, C. Puhrsch, G. Chauhan, S. Rao, A. Pagnoni, V. Boykis, A. Gonzales, and D. Eynard. Neurips large language model efficiency challenge: 1 llm + 1gpu + 1day, 2023.

[61] V. Shah, N. Ruiz, F. Cole, E. Lu, S. Lazebnik, Y. Li, and V. Jampani. Ziplora: Any subject in any style by effectively merging loras. arXiv preprint arXiv:2311.13600, 2023.

[62] M. Shen, S. Das, K. Greenewald, P. Sattigeri, G. Wornell, and S. Ghosh. Thermometer: Towards universal calibration for large language models. arXiv preprint arXiv:2403.08819, 2024.

[63] T. Shnitzer, A. Ou, M. Silva, K. Soule, Y. Sun, J. Solomon, N. Thompson, and M. Yurochkin. Large language model routing with benchmark datasets. arXiv preprint arXiv:2309.15789, 2023.

[64] S. P. Singh and M. Jaggi. Model fusion via optimal transport. Advances in Neural Information Processing Systems, 33:22045–22055, 2020.

[65] N. Slonim, Y. Bilu, C. Alzate, R. Bar-Haim, B. Bogin, F. Bonin, L. Choshen, E. Cohen-Karlik, L. Dankin, L. Edelstein, L. Ein-Dor, R. Friedman-Melamed, A. Gavron, A. Gera, M. Gleize, S. Gretz, D. Gutfreund, A. Halfon, D. Hershcovich, R. Hoory, Y. Hou, S. Hummel, M. Jacovi, C. Jochim, Y. Kantor, Y. Katz, D. Konopnicki, Z. Kons, L. Kotlerman, D. Krieger, D. Lahav, T. Lavee, R. Levy, N. Liberman, Y. Mass, A. Menczel, S. Mirkin, G. Moshkowich, S. Ofek-Koifman, M. Orbach, E. Rabinovich, R. Rinott, S. Shechtman, D. Sheinwald, E. Shnarch, I. Shnayderman, A. Soffer, A. Spector, B. Sznajder, A. Toledo, O. Toledo-Ronen, E. Venezian, and R. Aharonov. An autonomous debating system. Nature, 591:379 – 384, 2021. URL https://api.semanticscholar.org/CorpusID:232305184.

[66] S. U. Stich. Local SGD converges fast and communicates little. arXiv preprint arXiv:1805.09767, 2018.

[67] G. Stoica, D. Bolya, J. Bjorner, T. Hearn, and J. Hoffman. Zipit! merging models from different tasks without training. arXiv preprint arXiv:2305.03053, 2023.

[68] Y.-L. Sung, L. Li, K. Lin, Z. Gan, M. Bansal, and L. Wang. An empirical study of multimodal model merging. arXiv preprint arXiv:2304.14933, 2023.

[69] D. Tam, M. Bansal, and C. Raffel. Merging by matching models in task subspaces. arXiv preprint arXiv:2312.04339, 2023.

[70] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.

[71] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, et al. Gemma: Open models based on gemini research and technology. arXiv preprint arXiv:2403.08295, 2024.

[72] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[73] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni. Federated learning with matched averaging. arXiv preprint arXiv:2002.06440, 2020.

[74] H. Wang, F. M. Polo, Y. Sun, S. Kundu, E. Xing, and M. Yurochkin. Fusing models with complementary expertise. arXiv preprint arXiv:2310.01542, 2023.

[75] A. Warstadt, L. Choshen, A. Mueller, E. Wilcox, W. Adina, C. Zhuang, L. Tal, and R. Cotrerell. Findings of the BabyLM Challenge: Sample-efficient pretraining on developmentally plausible corpora. In Proceedings of the BabyLM Challenge. Association for Computational Linguistics (ACL), December 2023.

[76] A. Warstadt, A. Mueller, L. Choshen, E. Wilcox, C. Zhuang, J. Ciro, R. Mosquera, B. Paranjabe, A. Williams, T. Linzen, and R. Cotterell, editors. Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning, Singapore, Dec. 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.conll-babylm.0.

[77] S. Welleck, X. Lu, P. West, F. Brahman, T. Shen, D. Khashabi, and Y. Choi. Generating sequences by learning to self-correct. arXiv preprint arXiv:2211.00053, 2022.

[78] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, and L. Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In International Conference on Machine Learning, 2022.

[79] P. Yadav and M. Bansal. Exclusive supermask subnetwork training for continual learning. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, Findings of the Association for Computational Linguistics: ACL 2023, pages 569–587, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.36. URL https://aclanthology.org/2023.findings-acl.36.

[80] P. Yadav, L. Choshen, C. Raffel, and M. Bansal. Compeft: Compression for communicating parameter efficient updates via sparsification and quantization, 2023.

[81] P. Yadav, Q. Sun, H. Ding, X. Li, D. Zhang, M. Tan, P. Bhatia, X. Ma, R. Nallapati, M. K. Ramanathan, M. Bansal, and B. Xiang. Exploring continual learning for code generation models. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 782–792, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.68. URL https://aclanthology.org/2023.acl-short.68.

[82] P. Yadav, D. Tam, L. Choshen, C. Raffel, and M. Bansal. Resolving interference when merging models. ArXiv, abs/2306.01708, 2023. URL https://api.semanticscholar.org/CorpusID:259064039.

[83] P. Yadav, D. Tam, L. Choshen, C. Raffel, and M. Bansal. TIES-merging: Resolving interference when merging models. In Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

[84] M. Yamada, T. Yamashita, S. Yamaguchi, and D. Chijiwa. Revisiting permutation symmetry for merging models between different datasets. arXiv preprint arXiv:2306.05641, 2023.

[85] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, and N. Hoang. Statistical model aggregation via parameter matching. Advances in neural information processing systems, 32, 2019.

[86] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni. Bayesian nonparametric federated learning of neural networks. In International conference on machine learning, pages 7252–7261. PMLR, 2019.

[87] J. Zhu, A. Guha, D. Do, M. Xu, X. Nguyen, and D. Zhao. Functional optimal transport: map estimation and domain adaptation for functional data, 2021.

[88] J. Zhu, J. Qiu, A. Guha, Z. Yang, X. Nguyen, B. Li, and D. Zhao. Interpolation for robust learning: Data augmentation on wasserstein geodesics, 2023.

[89] J. Zhu, K. Greenewald, K. Nadjahi, H. S. d. O. Borde, R. B. Gabrielsson, L. Choshen, M. Ghassemi, M. Yurochkin, and J. Solomon. Asymmetry in low-rank adapters of foundation models. arXiv preprint arXiv:2402.16842, 2024.