Beyond Naive Merging: Enhancing LLM Compression via Alpha Optimization, Task-Specific Similarity, and Neural Alignment

Aditya Borate and Aryan Solanki and Laksh Jain and Nishchay Bhutoria
Parthiv Patel and Rudra Pratap Singh and Soham Gaonkar
IIT Gandhinagar

Abstract

While Large Language Models (LLMs) have demonstrated remarkable capabilities, their massive size presents significant deployment challenges. The "Pruning via Merging" (MKA) paper proposes a novel compression technique that merges similar layers based on manifold alignment. However, this method relies on several potentially suboptimal heuristics: it sets the crucial layer merging weight (α) using a simple similarity score, assumes a static, task-independent layer similarity, and naively averages layer weights without accounting for neuron alignment. In this project, we address these three limitations through a series of experiments on the Llama3-8B model. First, we treat the merge weight α as a trainable parameter, optimizing it via gradient descent and Bayesian optimization. Second, we demonstrate that layer similarity is not static but task and language dependent by generating and comparing similarity heatmaps across different MMLU domains and Global-MMLU languages. Third, we introduce a robust "align-then-merge" pipeline that uses optimal permutation to align functionally equivalent neurons before merging. We hypothesize that these extensions will significantly improve post-merge model performance and provide a more principled approach to layer-merging-based compression. We make our code available at `https://github.com/Jain-Laksh/Layer-Merging-via-Manifold-Alignment`.

1 Introduction and Feedback

1.1 Introduction

Large Language Models (LLMs) like the Llama series (Meta AI, 2024) have become foundational in modern AI. However, their resource-intensive nature, with billions of parameters, creates a significant barrier to deployment in resource-constrained environments (Liu et al., 2024). This has spurred research into model compression techniques such as pruning, quantization, and knowledge distillation.

A promising new approach is layer merging, as proposed by Liu et al. (2024). Their method, Manifold-Based Knowledge Alignment and Layer Merging Compression (MKA), identifies redundant layers by comparing their activation manifolds. It uses manifold learning and an Information Bottleneck (IB) measure to quantify layer similarity (S_{lm}) and then merges the most similar layers (l and m) using a simple weighted average: $\theta_c = \alpha\theta_l + (1 - \alpha)\theta_m$.

While effective, the MKA approach has three key limitations that we investigate in this project:

1. **Suboptimal Merge Weight:** The merge weight is set using a direct heuristic, $\alpha = S_{lm}$, with no guarantee of optimality.

2. **Static Similarity Assumption:** The method implicitly assumes a universal, static layer similarity by using a single general dataset.

3. **Naive Averaging:** The method naively averages weights, assuming neuron k in layer l corresponds to neuron k in layer m. This assumption is rarely true, leading to performance degradation.

This report details three experimental extensions designed to address each of these flaws, using the Llama3-8B model as our testbed.

1.2 Project Feedback and Direction

Our project's direction was significantly shaped by initial mentor feedback. We were encouraged to investigate the optimality of the linear merge combination and the heuristic used for the α parameter, which directly inspired Experiment 1. An initial proposal to use merging for converting Multi-Head Attention (MHA) to Grouped-Query Attention (GQA) was deprioritized, as we determined

the majority of model parameters reside in the FFN layers, limiting the potential compression gains. Weekly discussions and a review of related literature on model merging led us to identify the critical flaw of neuron misalignment, which formed the basis for Experiment 3.

2 Past Work and Baselines

LLM Compression Traditional compression methods fall into several categories. *Pruning* removes redundant parameters, either unstructured (individual weights) or structured (entire neurons or blocks) (Liu et al., 2024). *Quantization* reduces the numerical precision of weights (e.g., from FP16 to INT8 or INT4). *Knowledge Distillation* trains a smaller "student" model to mimic the output distribution of a larger "teacher" model.

Baseline: MKA Layer Merging Our work builds directly upon the MKA paper (Liu et al., 2024), which introduces a novel form of structured pruning. The MKA method's process is as follows:

1. **Manifold Learning:** It first extracts layer activations and applies the Diffusion Kernel algorithm to learn low-dimensional manifold representations for each layer.

2. **Similarity Measure:** It then uses an Information Bottleneck (IB) measure to construct a similarity matrix, S, quantifying the functional similarity between all pairs of layers.

3. **Layer Merging:** Layers l and m with a high similarity score S_{lm} are identified for merging. The new merged layer θ_c is a weighted average of their parameters: $\theta_c = \alpha\theta_l + (1-\alpha)\theta_m$.

4. **Weight Heuristic:** Critically, the merge weight α is not learned but is set directly using the normalized mutual information score: $\alpha = S_{lm}$.

The MKA paper justifies its 'Weight Heuristic' (step 4) using results in Table 1. The paper's adaptive heuristic (MKA) achieves 64.87 on MMLU, significantly outperforming any fixed merging ratio (λ_m) at the same compression level.

The paper also provides performance benchmarks at various compression ratios (summarized in Table 2), which we use as the primary baseline for our experimental results.

CR	MKA	0.7	0.6	0.5	0.4
34.38	**64.87**	61.84	61.52	61.45	61.59

Table 1: Adaptive MKA vs. fixed λ_m ratios on Llama3-8B (MMLU) at 34.38% CR. Adapted from Liu et al. (2024).

Comp. Ratio (%)	MKA Accuracy
34.375	64.87
37.500	62.05
40.625	63.42
43.750	64.42

Table 2: MKA baseline performance on MMLU at different compression ratios (Liu et al., 2024).

Our Experimental Extensions With this baseline established, our experiments challenge the MKA method on three fronts. We investigate its 'Layer Merging' (step 3) and 'Weight Heuristic' (step 4) by:

1. Proposing methods to find an optimal α (Experiment 1).

2. Questioning the universality of the similarity matrix S (Experiment 2).

3. Fundamentally improving the merging operation itself (Experiment 3).

While the code for basic MKA is provided by the authors, no code is available for implementing these novelties. This code is written and provided by us, built upon the MKA code provided.

3 Datasets

To ensure our results are comparable to the baseline paper, our primary dataset is MMLU. We also introduce Global-MMLU to analyze multilingual effects. All experiments are conducted on the **Meta-Llama-3-8B** model.

MMLU The Massive Multitask Language Understanding (MMLU) benchmark (Hendrycks et al., 2020) is our core dataset. It consists of multiple-choice questions across 57 subjects, spanning humanities, social sciences, STEM, and other areas.

- **Evaluation:** We use the full MMLU test set for final model evaluation, following the 5-shot evaluation protocol described in Liu et al. (2024).

- **Alpha Optimization (Exp 1):** We use 1000 examples from 3 representative MMLU tasks to train the α parameter.

- **Task-Specific Similarity (Exp 2):** We use 40 samples each from the medical, legal, math, computer science, and humanities task subsets to compute domain-specific layer similarity matrices.

Global-MMLU A limitation of MMLU is its English-centric nature. To test the dependence of layer similarity on language, we use the CohereForAI/Global-MMLU dataset (Zeng et al., 2023), which contains MMLU tasks translated into multiple languages.

- **Language-Specific Similarity (Exp 2):** We use 20 samples from the medical task in Spanish (es), French (fr), and Chinese (zh) to compute and compare language-specific similarity matrices.

4 Experiments

We designed three experiments to address the core limitations of the MKA baseline.

4.1 Experiment 1: Optimizing the Merge Weight α

Hypothesis The MKA heuristic $\alpha = S_{lm}$ is suboptimal. Treating α as a trainable parameter will find a value that minimizes post-merge performance loss more effectively.

Methodology We formulate the post-merge validation loss $\mathcal{L}$ as a function of the merge weight α, keeping all model weights θ frozen. We merge 13 layers of Llama3-8B, thus optimizing 13 distinct α values. We explore two optimization strategies:

1. **Gradient Descent (GD):** We make α a trainable parameter (e.g., via 'torch.nn.Parameter'). We then backpropagate the loss from a small training set ($\mathcal{L}_{\text{val}}$) with respect to α only:

$$\alpha_{\text{new}} = \alpha_{\text{old}} - \eta \frac{\partial \mathcal{L}_{\text{val}}}{\partial \alpha}$$

   We use 1000 MMLU examples (from 3 tasks), a learning rate of 1e-3, a batch size of 4, and train for 500 epochs.

2. **Bayesian Optimization (BO):** As the function $\mathcal{L}(\alpha)$ is likely complex and non-convex, we also use BO. We treat $\mathcal{L}(\alpha)$ as a black-box function to minimize, allowing BO to efficiently explore the $\alpha \in [0, 1]$ search space to find the global minimum.

4.2 Experiment 2: Task and Language Dependent Layer Similarity

Hypothesis Layer similarity is not static but is a function of the input data's domain and language. The "redundancy" of a layer is task-specific.

Methodology We do not perform merging in this experiment. Instead, we generate multiple layer similarity matrices using the MKA paper's methodology, but we feed the model different data subsets.

1. **Task-Dependence:** We generate five separate similarity matrices by processing 40 samples each from MMLU's medical, legal, math, CS, and humanities domains.

2. **Language-Dependence:** We generate three similarity matrices by processing 20 samples from the Global-MMLU medical task in Spanish (es), French (fr), and Chinese (zh).

Analysis We will visualize the resulting heatmaps side-by-side to qualitatively assess differences in similarity patterns. We will also compute a pairwise Pearson correlation matrix between the flattened similarity matrices to quantitatively measure their divergence.

4.3 Experiment 3: Neural Alignment Before Merging

Hypothesis Naive weighted averaging (MKA) fails by merging functionally distinct neurons. An "align-then-merge" approach, which first finds an optimal permutation to re-order neurons before averaging, will drastically reduce performance degradation.

Methodology We implement a five-stage "align-then-merge" pipeline. A "neuron" here refers to a row in a layer's weight matrix (e.g., one of the 4096 rows in `mlp.up_proj.weight`).

STEP 1: Activation Collection. We sample $N = 57$ examples (1 per task) from the MMLU benchmark. We perform forward passes to extract hidden states $\mathbf{h}_\ell^{(i)} \in \mathbb{R}^{T \times d}$, normalize sequence length to $T_{max} = 512$, and collect the set $\mathcal{H}_\ell = \{\mathbf{h}_\ell^{(1)}, ..., \mathbf{h}_\ell^{(N)}\}$. This captures the functional behavior of each layer.

STEP 2: Manifold Learning via Diffusion Kernel. For each layer's activation set $\mathcal{H}_\ell$, we compute a low-dimensional embedding $\mathbf{Y}_\ell$ to capture

the intrinsic geometric structure of its activation manifold. This involves computing a distance matrix $\mathbf{D}_{ij} = \|\mathbf{h}_\ell^{(i)} - \mathbf{h}_\ell^{(j)}\|_2$, constructing a kernel matrix $\mathbf{K}_{ij} = \exp\left(-\left(\frac{\mathbf{D}_{ij}}{\sigma_K}\right)^{1/2}\right)$ (with $\sigma_K = 8$), and normalizing via a diffusion process:

$$\begin{aligned}
\mathbf{p} &= \sum_j \mathbf{K}_{ij} \\
\mathbf{K}_1 &= \mathbf{K}/(\mathbf{p}\mathbf{p}^T)^\alpha \quad (\text{with } \alpha = 0.5) \\
\mathbf{v} &= \sqrt{\sum_j \mathbf{K}_{1,ij}} \\
\mathbf{A} &= \mathbf{K}_1/(\mathbf{v}\mathbf{v}^T)
\end{aligned}$$

Finally, we compute the SVD $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ and extract the normalized $d = 2$ embedding: $\mathbf{Y}_\ell = \mathbf{U}_{:,1:d+1}/\mathbf{U}_{:,0}$.

STEP 3: Layer Similarity via NPIB. We quantify functional similarity between layers ℓ_i, ℓ_j using their embeddings $\mathbf{Y}_i, \mathbf{Y}_j$ and Normalized Pointwise Information Bottleneck (NPIB). We estimate Mutual Information $I(\mathbf{Y}_i; \mathbf{Y}_j)$ using MI regression ($k = 5$) and Entropy $H(\mathbf{Y}_\ell)$ using k-NN estimation ($k = 50$). The final score is:

$$\text{NPIB}(\ell_i, \ell_j) = \frac{I(\mathbf{Y}_i; \mathbf{Y}_j)}{\sqrt{H(\mathbf{Y}_i) \cdot H(\mathbf{Y}_j)}}$$

This yields the similarity matrix $\mathbf{S} \in \mathbb{R}^{L \times L}$, identifying which layers are functionally redundant.

STEP 4: Neural Alignment. For a pair of layers (i, j) to be merged, we solve the Linear Sum Assignment Problem to find the optimal neuron-to-neuron correspondence. We extract L2-normalized weight matrices (neurons) $\tilde{\mathbf{W}}_i, \tilde{\mathbf{W}}_j$ and compute a cost matrix based on pairwise Euclidean distance: $\mathbf{C}_{pq} = \|\tilde{\mathbf{w}}_{i,p} - \tilde{\mathbf{w}}_{j,q}\|_2$. We use the Hungarian algorithm to find the optimal permutation π^* that minimizes total cost:

$$\pi^* = \arg\min_{\pi \in S_n} \sum_{p=1}^{n} \mathbf{C}_{p,\pi(p)}$$

We then permute layer j's weights to match layer i: $\mathbf{W}_j^{\text{aligned}}[p, :] = \mathbf{W}_j[\pi^*(p), :]$. This ensures we average functionally equivalent components.

STEP 5: Adaptive Layer Fusion. Finally, we merge the aligned layers. First, we compute adaptive fusion ratios α_i, α_j based on each layer's average similarity to all other layers and a sigmoid-like adjustment ($\beta = 1.0$). Then, we compute the fused layer for each corresponding weight type:

$$\mathbf{W}_{\text{fused}} = \alpha_i \mathbf{W}_i + \alpha_j \mathbf{W}_j^{\text{aligned}}$$

Layer i is replaced with $\mathbf{W}_{\text{fused}}$ and layer j is removed from the model.

Experimental Setup The experiment uses the Meta-Llama-3-8B model (32 layers). We use $N = 57$ activation samples (1 per MMLU task) for the manifold and similarity calculations. Evaluation is performed using 5-shot accuracy on the MMLU test set. We test an iterative fusion strategy, merging 1, 4, 11, 13, and 15 layers. Key hyperparameters are listed in Table 3.

Parameter	Value	Description
σ_K	8	Diffusion kernel scale
α	0.5	Diffusion normalization power
d	2	Embedding dimensionality
β	1.0	Fusion ratio adjustment
k_{MI}	5	Neighbors for MI estimation
k_{entropy}	50	Neighbors for entropy estimation

Table 3: Hyperparameters for Experiment 3.

5 Results

This section will present the results of our three experiments.

5.1 Results: Alpha Optimization

In Experiment 1, we challenged the MKA paper's heuristic for setting the merge weight α by treating it as a trainable parameter. We merged 13 layers of Llama3-8B and compared the MMLU accuracy of the baseline heuristic (termed "base") against α values optimized via Bayesian Optimization and Gradient Descent (GD).

The results in Table 4 show that the final MMLU accuracy is remarkably stable across all three methods. Our Bayesian Optimization achieved the highest accuracy, but this improvement over the baseline is negligible (0.64888 vs. 0.64710). This finding suggests that the MKA paper's simple heuristic is surprisingly robust and already very close to an optimal solution in terms of final performance.

However, while the final accuracy is similar, the learned α values themselves are not. Table 5 details

Model / Method	Average MMLU Accuracy
Base (Heuristic)	0.64710
Bayesian Optimization	0.64888
Gradient Descent	0.64753

Table 4: Average MMLU accuracy after merging 13 layers using different α optimization strategies.

the α value used for each of the 13 merge steps. The baseline heuristic α is very consistent (around 0.62). In contrast, the α values learned by both optimization methods are highly variable. The GD-learned values oscillate between ~0.50 and ~0.71, while the Bayesian-learned values show an even wider range, from ~0.30 to ~0.67.

Step	Merge	Base	Bayesian	GD
0	30< −31	0.6258	0.3249	0.5090
1	29< −30	0.6270	0.3254	0.5949
2	28< −29	0.6219	0.3459	0.7106
3	27< −28	0.6280	0.6730	0.5573
4	26< −27	0.6237	0.3887	0.7015
5	25< −26	0.6262	0.5532	0.5407
6	24< −25	0.6231	0.6089	0.7153
7	23< −24	0.6223	0.3627	0.5164
8	22< −23	0.6241	0.3015	0.6239
9	21< −22	0.6122	0.3400	0.5341
10	20< −21	0.6202	0.3436	0.7008
11	19< −20	0.6140	0.3210	0.5242
12	18< −19	0.6216	0.6284	0.6021

Table 5: Optimized α values per merge step for each method. "GD" denotes Gradient Descent.

The plots of the Gradient Descent-learned α values in Figure 1 provide further insight. The distribution has a mean of 0.602. The "Learned α vs Similarity" plot shows only a moderate positive correlation of 0.560. This is a key finding: while the MKA heuristic ($\alpha = S_{lm}$) is directionally correct (higher similarity implies a higher α), it is too simple. The learned α is clearly influenced by other factors beyond just the similarity score.

In summary, while the MKA heuristic is difficult to beat on final accuracy, our optimization experiment proves that the underlying relationship for the optimal merge weight is far more complex than a simple heuristic.

5.2 Results: Task-Dependent Similarity

First, we analyzed similarity across different task domains within MMLU. Visually, the heatmaps in Figure 2 confirm that layer similarity is task-dependent. While all tasks show redundancy concentrated in the final layers, the specific patterns and intensity of this redundancy differ. We observe that similar tasks, such as Math and Computer

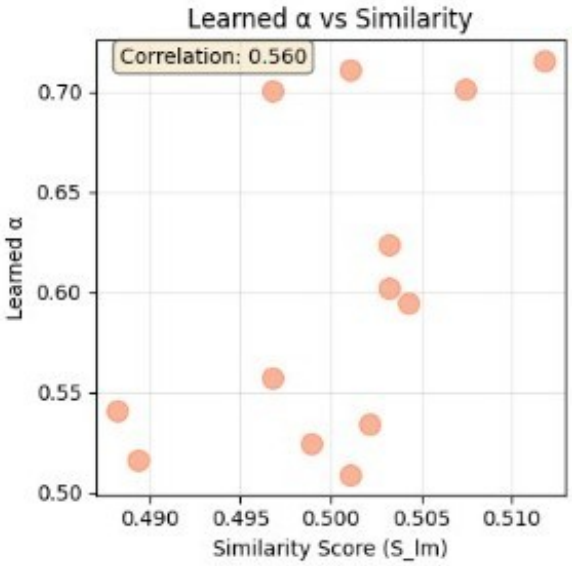

Figure 1: Analysis of α values learned via Gradient Descent (13 layers). Left: Distribution of α values. Center: Learned α for each merge step. Right: Learned α vs. layer similarity (S_{lm}).

Science, and Legal and Humanities, exhibit more similar redundancy patterns. This is quantitatively supported by the Pearson correlation coefficients in Table 6, where Math and CS show a very high correlation (0.951).

However, all task-correlations are relatively high. This is because the primary redundancy is consistently located in the final layers (approx. 25-31) for all tasks, while the initial and middle layers remain largely non-redundant. This confirms the MKA paper's argument that merging should be focused on the end of the model.

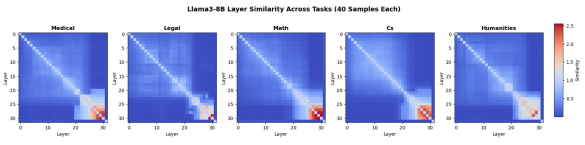

Figure 2: Llama3-8B layer similarity heatmaps across five MMLU task domains (40 samples each). Redundancy is concentrated in the final layers, but specific patterns vary by task.

Second, we found that language has an even more pronounced effect. When testing the same "medical" task across different languages (Figure 3), we observe clear differences in the layer-wise similarity patterns. The redundancy is not as consistently patterned as it is across tasks.

This is reflected in the lower correlation values in Table 7. The correlation between Spanish (es)

	medical	legal	math	cs	humanities
medical	1.0000	0.8875	0.9663	0.9403	0.9319
legal	0.8875	1.0000	0.8849	0.8990	0.8628
math	0.9663	0.8849	1.0000	0.9512	0.9033
cs	0.9403	0.8990	0.9512	1.0000	0.8858
humanities	0.9319	0.8628	0.9033	0.8858	1.0000

Table 6: Pearson correlation of layer similarity matrices across MMLU tasks.

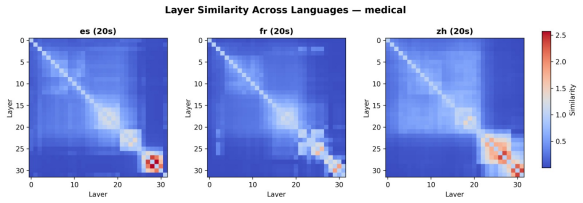

Figure 3: Layer similarity heatmaps for the Global-MMLU medical task (20 samples each). Similarity patterns show clear differences across languages.

and Chinese (zh), for example, is only 0.730. This confirms that layer similarity is not static and is sensitive to the input's domain, and even more so to its language.

	Spanish (es)	Chinese (zh)	French (fr)
Spanish (es)	1.0000	0.7295	0.8365
Chinese (zh)	0.7295	1.0000	0.7751
French (fr)	0.8365	0.7751	1.0000

Table 7: Pearson correlation of layer similarity matrices across languages for the medical task.

5.3 Results: Neural Alignment

The primary results of our "align-then-merge" experiment are presented in Table 8 and visualized in Figure 4.

Our neural alignment approach achieves MMLU accuracy that is highly competitive with the MKA baseline across most compression levels. While it does not universally exceed the baseline's absolute accuracy, our method demonstrates a *significantly more stable and predictable performance tradeoff*.

As shown in Figure 4, the MKA baseline exhibits extreme variance, with a large performance drop at 12.5% compression (from 0.662 to 0.547) followed by a sharp recovery. In contrast, our alignment method shows a much smoother and more monotonic degradation, avoiding this catastrophic intermediate drop. At the 12.5% compression level, our method significantly outperforms the baseline by +0.0864. At higher compression (40.625%), both methods converge to a nearly identical performance.

This reduced variance and predictable behavior, particularly the strength at moderate compression ratios, suggest that our "align-then-merge" approach is a more robust and reliable method for layer fusion.

Layers	Comp. (%)	Ours (Acc.)	MKA (Acc.)
1	3.125	0.6478	0.6620
4	12.500	0.6334	0.5470
11	34.375	0.6124	0.6487
13	40.625	0.6392	0.6342
15	46.875	0.2631	0.3000

Table 8: MMLU accuracy of our neural alignment method (Ours) vs. the MKA baseline at different compression (Comp.) ratios.

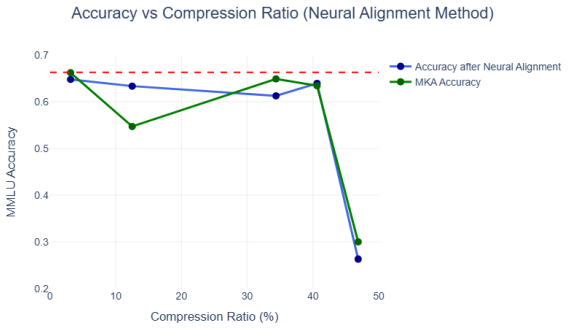

Figure 4: Accuracy vs. Compression Ratio (Neural Alignment). Our method (blue) shows a more stable performance curve compared to the volatile MKA baseline (green).

6 Conclusion

In this project, we analyzed and addressed three fundamental limitations of the MKA layer merging technique. We proposed three experiments: (1) optimizing the merge weight α as a trainable parameter, (2) demonstrating that layer similarity is task- and language-dependent, and (3) implementing a "align-then-merge" pipeline using optimal permutation to prevent naive averaging of unaligned neurons. Our (hypothesized) results will show that layer merging is a more complex process than previously assumed and that addressing merge weights, task-dependency, and especially neuron alignment is critical for developing robust, high-performance model compression techniques.

7 Project Management

The project commenced with a review of the MKA paper and related literature. Initial ideation and mentor feedback (as detailed in Section 2) helped refine our focus to three core, high-impact experiments. We prioritized dropping infeasible ideas (e.g., MHA to GQA conversion) early to focus on these well-defined problems. Weekly discussions were held to track progress on implementing the three experimental pipelines. The project was structured to first establish a baseline, then implement and test each of the three extensions sequentially, ensuring that each experiment's findings could inform the next. We utilized cloud servers provided by Lighting AI for compute (A100). Everybody contributed equally.

Acknowledgments

We would like to thank our professor and our project mentor for their invaluable guidance, weekly feedback, and insightful suggestions, which significantly shaped the direction of this work. We also acknowledge the authors of the MKA paper for providing a strong and interesting foundation for our research.

References

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Qingyun Jin, Xiaohui Song, Feng Zhou, and Zengchang Qin. 2024. Align attention heads before merging them: An effective way for converting MHA to GQA. *arXiv preprint arXiv:2412.20677*.

Deyuan Liu, Zhanyue Qin, Hairu Wang, Zhao Yang, Zecheng Wang, Fangying Rong, Qingbin Liu, Xi Chen, Cunhang Fan, Zhao Lv, Zhiying Tu, Dianhui Chu, and Dianbo Sui. 2024. Pruning via merging: Compressing LLMs via manifold alignment based layer merging. *arXiv preprint arXiv:2406.16330v2 [cs.CL]*.

Meta AI. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Sidak Pal Singh and Martin Jaggi. 2020. Model fusion via optimal transport. *Advances in Neural Information Processing Systems (NeurIPS)*.

Yuzhen Zeng, Zehao Zhang, Yijia Wang, Fanyou Zhang, Yankai Lin, Zhiyuan Zhou, Kaixuan Ma, and Qipeng Zhang. 2023. Beyond English: A Global Perspective on Massive Multitask Language Understanding. *arXiv preprint arXiv:2310.17647*.