A Pipeline to Assess Merging Methods via Behavior and Internals

Yutaro Sigrist[*1] **and Andreas Waldis**[1,2]
[1]Information Systems Research Lab, Lucerne University of Applied Sciences and Arts
[2]Ubiquitous Knowledge Processing Lab (UKP Lab), Technical University of Darmstadt
www.hslu.ch www.ukp.tu-darmstadt.de

Abstract

Merging methods combine the weights of multiple language models (LMs) to leverage their capacities, such as for domain adaptation. While existing studies investigate merged models from a solely behavioral perspective, we offer the first comprehensive view by assessing and connecting their behavior and internals. We present a novel evaluation pipeline that first merges multiple parent LMs and then evaluates the merged models in comparison to the initial ones based on their behavior on downstream tasks, like MMLU, and the internal encoded linguistic competence. We showcase this pipeline by assessing the merging of instruction fine-tuned with math- and code-adapted LMs from the `Qwen2.5` family. Our results show that merging methods impacts behavior and internals differently. While the performance of merged models is typically between that of the two parent models, their encoded information about linguistic phenomena – particularly in *morphology* and *syntax* – can surpass the parent models. Moreover, we find weak ranking correlation between this behavior and internal evaluation. With our pipeline and initial results, we emphasize the need for more comprehensive evaluations of model merging methods to gain a faithful understanding of their capabilities and reliability, beyond potential superficial behavioral advances.

1 Introduction

With the rise of competitive open-weight models (Dubey et al., 2024; Jiang et al., 2024), adapting large language models (LLMs) to specific use cases has become common practice. One approach, fine-tuning, enables adaptation to particular domains, such as mathematics or coding (Lewkowycz et al., 2022). Still, it suffers from high computational costs and the risk of catastrophic forgetting (Luo et al., 2023). When a model must handle a variety of tasks, fine-tuning becomes more complex (Lu et al., 2024). Model merging offers methods that overcome some of these issues. These methods combine multiple models into a single one, thereby enhancing performance across the tasks for which the individual parent models were fine-tuned (Yang et al., 2024).

Existing work in model merging primarily focuses on techniques to improve the performance of merged models. For example, Lu et al. (2024) explored strategies for merging domain-adapted parent models, such as those in materials science and engineering, to efficiently transfer this domain adaptation to other models. In addition, Dziadzio et al. (2024) introduces the concept of temporal model merging, addressing the challenge of integrating knowledge from multiple parent models trained on various tasks over time. Furthermore, Goddard et al. (2025) introduces MergeKit, a merging toolkit supporting various methods.

While this variety of studies highlights the popularity of model merging, existing work focuses primarily on assessing these models based on their behavior. As a result, we lack a comprehensive evaluation that assesses and connects both model behavior and internals. This research gap means we risk relying on potentially unstable merged models based on their seemingly better outputs. With this work, we address this shortage in model merging research and ask the following question:

> *How does model merging affect the internal representations of language models?*

To answer this, we present an evaluation pipeline (section 3) with three stages: **1)** we merge multiple parent models with different strategies, **2)** we evaluate both the behavior and internals of the parent and resulting merged models, and **3)** we connect the findings from both evaluations. We showcase this pipeline through experiments on the *Qwen2.5* model family (Qwen et al., 2025), which merges

*Corresponding author yutarosigrist@gmail.com

Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pages 307–316
November 9, 2025 ©2025 Association for Computational Linguistics

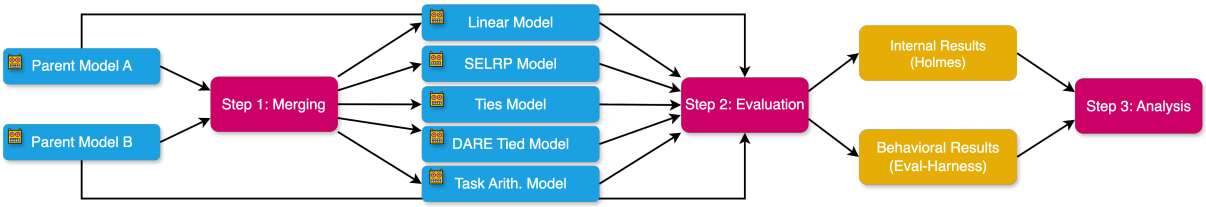

Figure 1: This flowchart shows a model evaluation pipeline. Input "Instruct" and "Math/Coder" models undergo a "Merging" process (using techniques like `Linear`, `SLERP`, `TIES`, `DARE TIES`, `Task Arithmetic`). The merged output is then evaluated by "Holmes-Evaluation (Flash-Holmes)" for linguistic competencies (reasoning, morphology, etc.) and "Evaluation LM-Harness" using benchmarks (`Leaderboard`). The pipeline concludes with analyzing the correlation between Harness and Holmes.

instruction fine-tuned and domain-adapted models. Our results show that while merged models generally perform between the parent models, their encoded linguistic competence—particularly for *morphology* and *syntax*—can be stronger than the parent ones'. At the same time, weak rank correlations between behavior and internals underscore the need for comprehensive evaluations, as single perspectives are insufficient to capture the full complexity of language models (Hu and Levy, 2023; Waldis et al., 2025), for and beyond model merging.

With this work, we provide the ground for more comprehensively assessing model merging methods and resulting models by contributing as follows:

- We introduce the first pipeline that combines model merging with a comprehensive evaluation of model behavior and internals.
- We present initial insights into connecting the behavioral and internal interpretability perspectives in the context of domain adaptation scenarios.
- We make all code, including pipeline and analysis, online available.[1]

2 Related works

Model Merging Model merging enables the efficient combination of specialized models (Yang et al., 2024). This includes simpler methods like linearly averaging multiple parent models (Wortsman et al., 2022) or spherical interpolation (SLERP)[2]. More sophisticated methods try to locate task-specific regions to better preserve specific skills of the parent models, popular examples are *Task Arithmetic* (Ilharco et al., 2023), *Ties* (Yadav et al., 2023), or *Dare-Ties* (Yu et al., 2024). The general popularity of such methods is evident in the development of toolboxes that easily merge two or more parent models, such as *MergeKit* (Goddard et al., 2024), resulting in thousands of merged models available on Huggingface.

Model Evaluation A predominant line of research focuses on evaluating language models based on their behavior. This includes general language understanding (GLUE; Wang et al. 2018 or SuperGLUE; Wang et al. 2019), question answering (SQuAD; Rajpurkar et al. 2016), or more broad evaluations, as done in HELM (Liang et al., 2023) or *evaluation harness* (Gao et al., 2024). Moreover, research also focused on comprehensively assessing specific domains, like factuality (Chen et al., 2023; Muhlgay et al., 2024), medical texts (Bedi et al., 2025), or legal reasoning (Guha et al., 2023). While these benchmarks focus on model behavior, there has been little comprehensive work addressing model internals. Conneau et al. (2018) introduce a benchmark with ten tasks to assess linguistic properties of sentence representations, Warstadt et al. (2020) presents a collection of minimal pairs to examine whether language models can internally differentiate between linguistic acceptable and unacceptable sentences, and Waldis et al. (2024) introduced, with `Holmes`, a benchmark that studies the linguistic competence of language models based on their internals across five linguistic phenomena and more than 160 distinct probing tasks.

With this work, we extend the evaluation scope of merging methods to model internals, a previously understudied aspect in model merging methods, in favor of a more comprehensive understanding of how these approaches combine model

[1] https://github.com/yusigrist/LLM-Merging-Piepline
[2] https://github.com/Digitous/LLM-SLERP-Merge

weights, and offer the base to assess how this changes information flow within models.

3 The Pipeline

With the presented pipeline (Figure 1), we offer a flexible workflow to merge parent models, and evaluate these parents as well as the resulting merged models from a behavioral and internal perspective. Finally, we connected these distinct interpretability perspectives to gain a more in-depth understanding of model workings.

3.1 Merging methods

In a first step, this pipeline uses *MergeKit* (Goddard et al., 2024) to combine two parent models. Table 1 provides an overview of these methods, which vary in complexity, and compares them based on a few essential features. Note that performance always depends on the configuration and model, such that `Linear` can work better than `DARE TIES` under certain conditions.

Linear averages corresponding weights of multiple models, often fine-tuned from a common base with different hyperparameters. This method stands out because of its low complexity and very low difficulty when implemented with Mergekit, making it highly accessible to practitioners. The approach maintains low power usage during both inference and merging processes, while supporting multi-model combinations effectively. Key Application/Strength are simplicity, improved accuracy, and robustness without an increase in inference costs for similar models. Despite its simplicity, performance can vary significantly depending on configuration and, in some cases, may be limited compared to more complex methods (Wortsman et al., 2022).

SLERP interpolates weights between two models along spherical paths, with the aim of a smoother blend than `Linear`. This method operates with moderate complexity while maintaining low power consumption during both inference and merging. However, it is limited to only two models in practice, making it less suitable for multi-model scenarios. The implementation difficulty of Mergekit remains low, making it accessible for most applications. Key Application/Strength are effective two-model merging, potentially finding lower loss barrier paths. Performance typically exceeds linear methods when properly configured, offering a balanced approach between simplicity and effectiveness (Shoemake, 1985; Chris Alexiuk, 2024).

Task Arithmetic computes "task vectors" (fine-tuned base weights) and adds them to a base model to combine capabilities. This approach maintains moderate complexity while effectively supporting multi-model merging scenarios. The method benefits from low power usage during both inference and merging phases, with relatively low implementation difficulty in Mergekit frameworks. Key Application/Strength are flexible multitasking combination in model editing. Performance generally surpasses `Linear` approaches, providing a practical solution to combine diverse model capabilities without significant computational overhead (Ilharco et al., 2023).

TIES trims insignificant parameter changes, selects a dominant sign for conflicts, and merges only aligned parameters. This method introduces high complexity in its implementation, which requires careful configuration and understanding of the interaction of the parameters. Despite the increased complexity, it maintains low power usage during inference and merging, while supporting multi-model combinations. Key Application/Strength are the reduction of parameter interference, especially for sign conflicts, in multi-model merging. Performance typically exceeds `SLERP` methods when properly implemented, making the additional complexity worthwhile for demanding applications (Yadav et al., 2023).

DARE TIES uses the DARE (Drop And REscale) scheme to sparsify task vectors (randomly drops and rescales parameters) before applying `TIES`-merging. This advanced method operates with high complexity, building on the TIES framework with additional parameter sparsification techniques that require expertise in both `DARE and TIES` methodologies. Power consumption remains low during both inference and merging, while supporting comprehensive multi-model integration. Key Application/Strength are the further mitigated interference by reducing parameter density before `TIES`. Performance generally surpasses `SLERP` methods and can exceed `TIES` in many configurations, representing a state-of-the-art approach to model merging (Yadav et al., 2023; Yu et al., 2024).

Feature	Linear	SLERP	Task Arithmetic	TIES	DARE TIES
Complexity	low	moderate	moderate	moderate	high
Power usage (Inference)	low	low	low	low	low
Power usage (Merging)	low	low	low	low	low
Multi-Model	yes	2 models	yes	yes	yes
Performance	simple	>Linear	>Linear	>SLERP	>SLERP
Mergekit Difficulty	very low	low	low	moderate	high

Table 1: Comparison of a set of Features for five different merging methods (Linear, SLERP, Task Arithmetic, TIES, DARE TIES). The Performance always depends on the configuration and model. So in some cases Linear can be better than DARE TIES.

3.2 Evaluation

In a second step, our pipeline automatically evaluates both the parent models and the resulting merged ones. This includes a behavioral evaluation using the eval-harness library (Gao et al., 2024) and the streamlined and efficient version of Holmes (Waldis et al., 2024), which comprehensively assesses information about linguistic phenomena within model internals.

3.2.1 LM-Harness-Evaluation

The LM-Harness-Evaluation is a comprehensive evaluation framework to assess language models (Gao et al., 2024). It includes a variety of tasks in favor of a standardized and reproducible evaluation of these models. For our study, we use the following tasks also used in the OpenLLM evaluation leaderboard (Fourrier et al., 2024):

- BBH: Collection of LLM tasks across domains, for example, language understanding, mathematical reasoning, common sense, and world knowledge.
- math hard: High school level competitions for mathematical problems: complex algebra, geometry problems, etc.
- MUSR: Reasoning on and understanding of long texts: language understanding, reasoning capabilities
- GPQA: PhD-level knowledge multiple choice questions in science: Chemistry, Biology, and Physics
- MMLU-PRO: Expert reviewed multiple choice questions across domains, for example: medicine and healthcare, law and ethics, engineering, mathematics

3.2.2 Holmes-Evaluation

Using Holmes, we evaluate the internal representations using classifier based probing (Belinkov, 2022) Specifically, we use Flash-Holmes, a streamlined and efficient version of the benchmark that preserves the effectiveness of the evaluation, even substantially reducing the number of instances. It assesses a range of linguistic phenomena, including morphology, syntax, semantics, reasoning, and discourse, using more than 160 unique probing tasks. This involves training simple linear classifiers on the internal representations of the models' last layer to predict specific linguistic properties, thereby assessing how well the model encodes this information (Hewitt and Liang, 2019; Voita and Titov, 2020; Waldis et al., 2024).

4 Evaluation Results

We merge Instruct with Coder and Math models from the 7B Qwen-2.5 family (Qwen et al., 2025). Leveraging both the LM-Harness and Holmes frameworks reveals a significant and consistent divergence in how model merging impacts external behavior versus internal representations. While downstream tasks, the best merged models perform between the two parent models, the inherent linguistic competence encoded within the model internals tends to increase, particularly when merging Instruct and Math.

4.1 Model Behavioral Evaluation

We first summarize behavioral results from the LM-Harness evaluation in Figure 2.

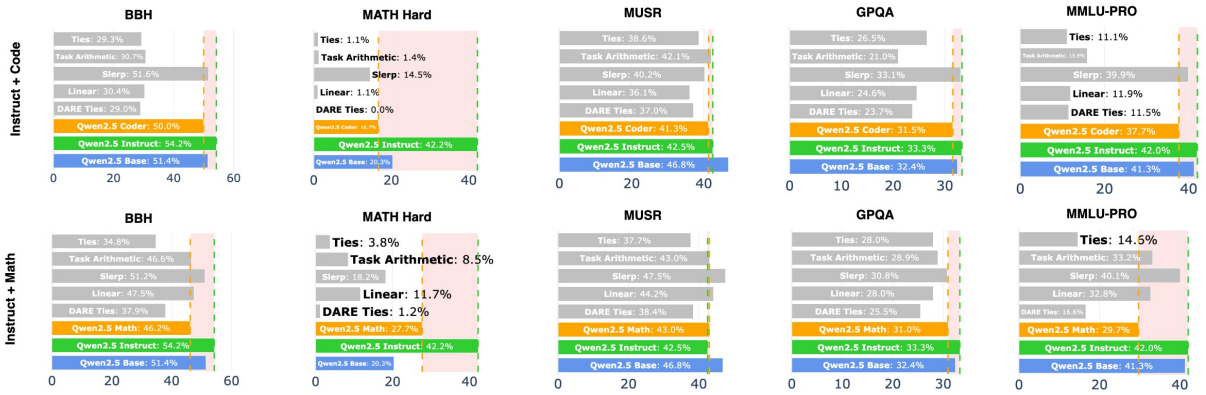

Figure 2: This horizontally grouped bar chart illustrates the absolute performance of each model across the main `Harness` tasks. The red shaded area highlights the performance gap between the instruct and math/coder models. Notably, simpler methods generally outperform more complex ones. Overall, all merged models show poorer performance compared to their parent models.

The behavioral compromise of model merging. These results consistently demonstrate that merged models can not match the performance of one of their parent models, but mostly perform between them. As an exception, in both `Instruct + Math` and `Instruct + Coder` experiments, the merged models fail to reach at least one parent model's performance for `MATH Hard`. These results align with the general low performance of domain-adapted models and previous results, which underscore the importance of instruction-tuning in combination with domain adaptation (Beeching et al., 2023).

Simple model merging methods excel. Among the merging methods, a clear hierarchy emerges. Simpler methods consistently outperform more complex ones. As shown in Figures 4 and 3, `SLERP` stands out as the most effective method, frequently achieving the highest scores among the merged models and having the largest number of subtasks where its performance is "better" than both parents. `Linear` and `Task Arithmetic` follow, typically performing "between" the two parent models. In stark contrast, the more sophisticated methods, `TIES` and `DARE TIES`, which are designed to mitigate parameter interference, consistently got the poorest results. Their performance is often categorized as "worse" than both parent models, suggesting that their approach to resolving weight conflicts may be detrimental to the model's ability to perform complex, multi-step tasks.

Figure 3: This stacked bar chart illustrates how each model's performance on `Leaderboard` Subtasks compares to both the `Instruct and Coder` models. Generally, most models perform either between or worse than these two baselines. However, simpler merging methods show some subtasks where they perform better."

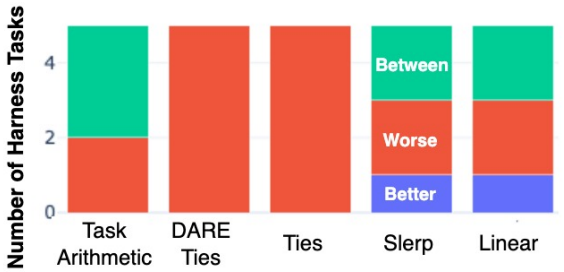

Figure 4: This stacked bar chart illustrates how each model's performance on `Leaderboard` Subtasks compares to both the `Instruct and Math` models. Generally, most models perform either between or worse than these two baselines. However, simpler merging methods show some subtasks where they perform better."

4.2 Model Internal Evaluation

We show in Figure 5, 7, and 6 the result of evaluating model internals regarding their inherent en-

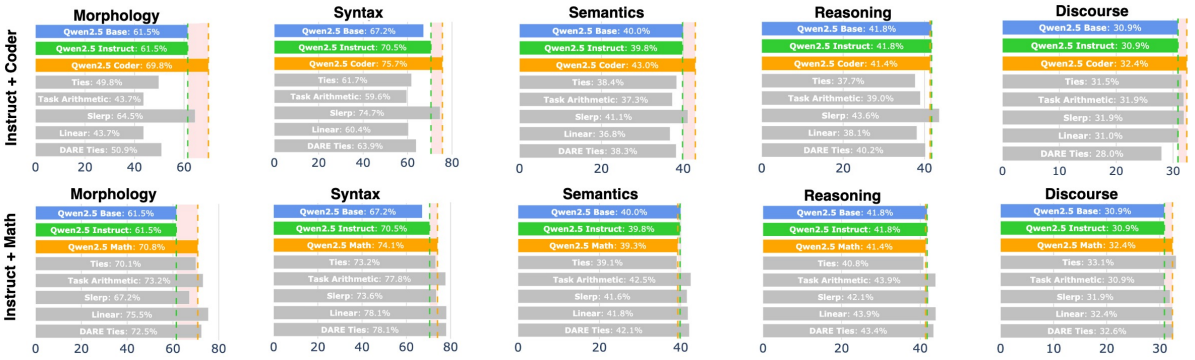

Figure 5: This horizontally grouped bar chart illustrates the absolute performance of each model across the linguistic competencies. The red-shaded area highlights the performance gap between the `Instruct` and `Math/Coder` models. Notably, merged models mostly outperform or perform similarly to the parent models.

coded linguistic competence.

More encoded information with more training. More generally, we found that both instruction tuning (`Instruct`) and domain adaptation (`Coder` and `Math`) result in more information about linguistic competence compared to the `Base` model from which all originate. This effect is particularly evident for domain adaptation, where we assume that the larger amount of data than used for instruction tuning allows LMs to capture more information about linguistic phenomena.

Model merging increases encoded information. Next, we focus on how model merging affects the model internals and find that the impact of combining models diverges from behavioral evaluations. Notably, we found that information about linguistic phenomena can increase when models are merged. This effect is most pronounced for **`morphology` (word structure) and `syntax` (sentence structure)**. This suggests that merging can effectively combine the complementary structural knowledge from both models, resulting in internal representations with more information.

More information when using simpler merging methods. In the next step, we compare the different merging methods regarding the information encoded in the resulting models. Similarly, to the behavioral evaluation, we find that simpler methods (`Liner` or `SLERP`) generally preserve more information than more complex methods. This effect is not as pronounced as when evaluating model behavior. Notably, we find the most difference among models (parent and merged ones) in more formal phenomena, like `syntax` and `morphology`. However, there are fewer differences in phenomena that

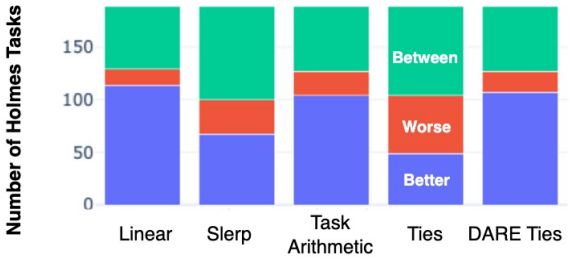

Figure 6: This stacked bar chart illustrates how each model's performance on `Holmes` tasks compares to both the `Instruct and Math` models. Generally, most models perform either better than or between these two baselines.

are intuitively linked to actively using language (`reasoning`, `semantics`, or `discourse`), which we often attribute to language models. This improvement for formal phenomena also suggests that merging can effectively transfer the information gain from seeing more tokens in further pretraining steps, as shown in (Waldis et al., 2024).

Adapted domain matters for merging compatibility. We compare the influence of the particular domain with merging into the `Instruct` model and find that information gain from model merging is not uniform. It is more pronounced in `Instruct + Math` than in `Instruct + Coder` experiments. In the `Instruct + Math` experiment (Figure 6), nearly all merging methods produce models that outperform both the `Instruct` and `Math` models for these two phenomenon types. In contrast, we see that the `Coder` parent models have slightly richer model internals than the `Math` one. These insights suggest that model merging is sensitive to the specific domain in which it is applied. Specifically, we believe that the language used for `Math`

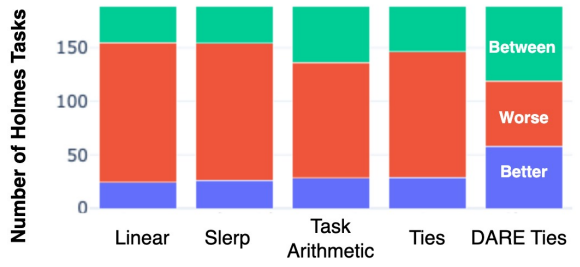

Figure 7: This stacked bar chart illustrates how each model's performance on `Holmes` tasks compares to both the `Instruct and Coder` models. Generally, most models perform either better than or between these two baselines.

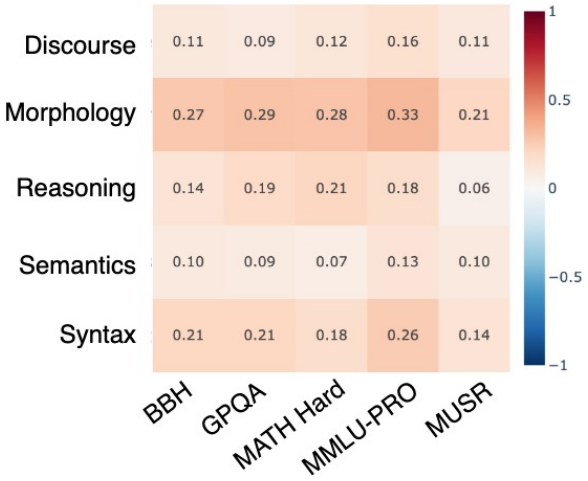

Figure 8: This plot shows a heatmap of the correlation between the linguistic competencies from `Holmes` and grouped `eval-harness` subtasks. It shows that `syntax` and `morphology` have the highest correlation with behavior.

adaptation is more similar to instruction tuning than that used to further pre-train the `Coder` models, resulting in better merging compatibility.

5 Discussion

Next, we discuss the connection results of the initial evaluation of model merging methods from a behavioral and internal perspective.

The efficacy of simple merging methods. In contrast to our expectations, we found that simple merging methods (`SLERP` and `Linear`) outperform more complex ones, such as `TIES` and `DARE TIES`, in both behavioral and internal evaluations. We hypothesize this is due to the **preservation of weight space geometry**. This is particularly evident for `SLERP`. We believe that finding the shortest path on the hypersphere of normalized weights respects the geometric relationships between parameters more faithfully. This geometric integrity appears to be essential for maintaining the functional coherence of the parent models. Moreover, this supremacy underlines the need for having more holistic evaluations to assess the advantages and limitations of merging methods comprehensivly. We see this work, with the presented pipeline and initial results, offering the first step in this direction.

Divergence between behavioral and internal evaluation of merging methods. The presented results indicate that behavioral and internal evaluations diverge substantially. While the behavioral performance of merged models decreases and is between that of the parents, the amount of encoded information within these models can increase beyond that of the parent models. This divergence highlights how internal and behavioral evaluation offer different perspectives and underscores the need for a better understanding of how these distinct interpretability perspectives interact. Thus, comprehensive methods and evaluations, as presented in this work, are not only indispensable for a better understanding of various model merging methods but also essential for a more general understanding of language models, beyond those that are merged.

Weak correlations of behavior and internals. Finally, empirically discuss results from model behavior and internals. We correlate the behavioral and internal results per model for tasks (eval-harness) and phenomena type (Holmes) in Figure 8. The correlations between linguistic competencies and downstream task performance are **weak to medium**. While we found the strongest correlation for `morphology` and `syntax` with *eval-harness* tasks, with Pearson Correlation values ranging from 0.14 to 0.33. These results underline, again, that **evaluating a model from a single perspective is insufficient**. A model that offers superior performance on a single leaderboard is not necessarily as internally rich as we might intuitively assume.

6 Conclusion

In this work, we introduce a novel evaluation pipeline that integrates model merging with model behavior and internal evaluations. With this novel methodology, we comprehensively assess the dynamics of different model merging methods.

Specifically, we focused on combining instruction fine-tuned models with math and coding adapted models. These initial results suggest clear divergence between model behavioral and internal evaluations. While merged models tend to perform between the two parent models, they can encode more linguistic information than these two models, particularly for `morphology and syntax`. Moreover, these results also suggest that simpler merging methods often outperform more complex ones, which underlines the necessity of comprehensive evaluation to better understand whether methods offer general superiority.

With these insights, we can directly answer our initially raised research questions: model merging affects internal representations, increasing the amount of information encoded in these representations beyond that of the parent, in a manner that differs from what behavioral evaluations suggest. With this divergence, we recognize that more fine-grained experiments are necessary to further study and strengthen the findings presented in this work, which are essential for gaining a deeper understanding and improving model merging methods.

Limitations

Model Layers In this study, we focus solely on the last layer of language models to investigate what information is encoded. While these would have expanded the scope of this work, studying how information flows through all model layers during model merging can further enhance our understanding of model merging methods, as well as of language models in general.

Language Models This study aims to present, through the pipeline, the methodological groundwork to assess model merging methods more comprehensively. In this context, we present initial results to showcase the effectiveness of this pipeline and derive first insights that guide investigations of model merging. For this purpose, we only experimented with models of one model family (QWEN-2.5). However, evaluating our results alongside those of other families could strengthen our findings and also uncover further differences among models.

English Language Given the widespread availability of evaluation resources, we limited this study to the English language. Thereby, the presented pipeline is not directly applicable to multilingual merging methods (Tao et al., 2024).

Acknowledgements

We thank the anonymous reviewer for their valuable feedback and discussions. Andreas Waldis is supported by the Hasler Foundation Grant No. 21024.

References

Suhana Bedi, Hejie Cui, Miguel Fuentes, Alyssa Unell, Michael Wornow, Juan M. Banda, Nikesh Kotecha, Timothy Keyes, Yifan Mai, Mert Oez, Hao Qiu, Shrey Jain, Leonardo Schettini, Mehr Kashyap, Jason Alan Fries, Akshay Swaminathan, Philip Chung, Fateme Nateghi, Asad Aali, Ashwin Nayak, Shivam Vedak, Sneha S. Jain, Birju S. Patel, Oluseyi Fayanju, Shreya J. Shah, Ethan Goh, Dong-han Yao, Brian Soetikno, Eduardo Pontes Reis, Sergios Gatidis, Vasu Divi, Robson Capasso, Rachna Saralkar, Chia-Chun Chiang, Jenelle A. Jindal, Tho Pham, Faraz Ghoddusi, Steven Lin, Albert S. Chiou, Christy Hong, Mohana Roy, Michael F. Gensheimer, Hinesh Patel, Kevin Schulman, Dev Dash, Danton Char, Lance Downing, François Grolleau, Kameron C. Black, Bethel Mieso, Aydin Zahedivash, Wen-wai Yim, Harshita Sharma, Tony Lee, Hannah Kirsch, Jennifer Lee, Nerissa Ambers, Carlene Lugtu, Aditya Sharma, Bilal Mawji, Alex Alekseyev, Vicky Zhou, Vikas Kakkar, Jarrod Helzer, Anurang Revri, Yair Bannett, Roxana Daneshjou, Jonathan H. Chen, Emily Alsentzer, Keith E. Morse, Nirmal Ravi, Nima Aghaeepour, Vanessa Kennedy, Akshay Chaudhari, Thomas Wang, Sanmi Koyejo, Matthew P. Lungren, Eric Horvitz, Percy Liang, Mike Pfeffer, and Nigam H. Shah. 2025. Medhelm: Holistic evaluation of large language models for medical tasks. *CoRR*, abs/2505.23802.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023. FELM: Benchmarking factuality evaluation of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Annie Surla Chris Alexiuk. 2024. [link].

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Sebastian Dziadzio, Vishaal Udandarao, Karsten Roth, Ameya Prabhu, Zeynep Akata, Samuel Albanie, and Matthias Bethge. 2024. How to merge your multimodal models over time?

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open llm leaderboard v2. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2025. Arcee's mergekit: A toolkit for merging large language models.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's MergeKit: A toolkit for merging large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485, Miami, Florida, US. Association for Computational Linguistics.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, K. Aditya, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John J. Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael A. Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks.

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5040–5060, Singapore. Association for Computational Linguistics.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian

Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yüksekgönül, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Trans. Mach. Learn. Res.*, 2023.

Wei Lu, Rachel K. Luu, and Markus J. Buehler. 2024. Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *CoRR*, abs/2308.08747.

Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2024. Generating benchmarks for factuality evaluation of language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 49–66. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ken Shoemake. 1985. Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, page 245–254, New York, NY, USA. Association for Computing Machinery.

Mingxu Tao, Chen Zhang, Quzhe Huang, Tianyao Ma, Songfang Huang, Dongyan Zhao, and Yansong Feng. 2024. Unlocking the potential of model merging for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8705–8720, Miami, Florida, USA. Association for Computational Linguistics.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length.

Andreas Waldis, Vagrant Gautam, Anne Lauscher, Dietrich Klakow, and Iryna Gurevych. 2025. Aligned probing: Relating toxic behavior and model internals. *CoRR*, abs/2503.13390.

Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych. 2024. Holmes: A benchmark to assess the linguistic competence of language models.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models.

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

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch.