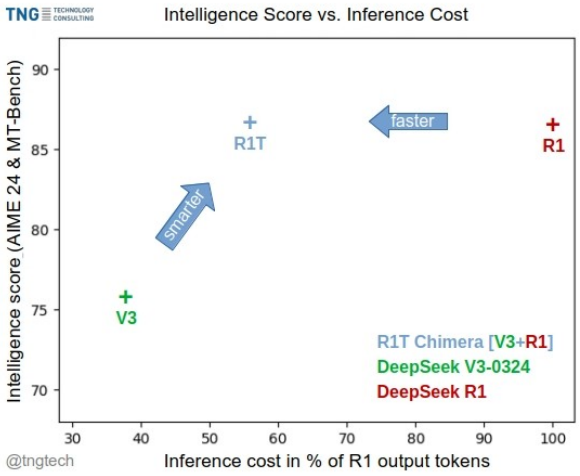
Assembly of Experts: Linear-time construction of the Chimera LLM variants with emergent and adaptable behaviors

Henrik Klagges[1,*], Robert Dahlke[1,*], Fabian Klemm[1], Benjamin Merkel[1], Daniel Klingmann[1], David A. Reiss[1], and Dan Zecha[1]

[1]TNG Technology Consulting GmbH, Germany
[*]joint first authorship

research@tngtech.com

31 May 2025



Abstract

Requiring 10^{13}–10^{15} FLOPs to calculate one 8 bit weight in an LLM during pretraining is extremely expensive and seems inefficient. To better leverage the huge investments made into pretrained models, we develop the new "Assembly-of-Experts" (AoE) construction method to create capable child variants of existing Mixture-of-Experts parent models in linear time. Model weight tensors get interpolated individually, allowing to enhance or suppress semantic features of the parents.

Varying the proportion of weights taken from the parent models, we observe some properties of the AoE child model changing gradually, while other behavioral traits emerge with a sharp transition. Surprisingly, nearly every generated model is functional and capable, which makes searching the model space straightforward.

We construct the DeepSeek R1T "Chimera", a 671B open-weights hybrid model combining DeepSeek's V3-0324 and R1 model variants. The child inherits only the routed expert tensors of R1, but still achieves about R1-level intelligence. At the same time, it uses about 40% fewer output tokens, close to V3 speed. Constructed without any fine-tuning or distillation, the Chimera exhibits surprisingly compact, orderly reasoning compared to its parent models.

1 Introduction

Assembling large language models (LLMs) by merging parameters of several parent models disrupts the costly paradigm of gradient-based fine-tuning and enables a lightweight composition of new models. Conventional adaptation methods like instruction fine-tuning and RLHF are very effective, but require expensive gradient updates and extensive training data. Creating new models by selecting or interpolating existing parameters from parent models is an additional method that can be performed at a fraction of the computational cost, at the same time resulting in child models that inherit desirable characteristics from each parent [1, 2, 3, 4, 5, 6, 7, 8, 9].

In this work, we demonstrate the usefulness of assembling large-scale Mixture-of-Expert (MoE) models by merging structural components of the model with individually chosen merging configurations to steer model behaviours. Prior work has shown promising results in merging mid-sized (7B to 14B) models, but identified "significant challenges in simultaneously maintaining reasoning performance while substantially reducing response length" ([10]) for models larger than 32B parameters. We make use of the fine-granular structure of large MoE architectures and extend the regular model merging methodology to assembling models. With examples from the DeepSeek V3 family of 671B-parameter models, we demonstrate that the assembly process remains feasible and effective, even at this large parameter scale.

One of our goals is to assemble models that inherit desirable capabilities from their parents – such as combining the advanced reasoning capabilities of DeepSeek-R1 with the instruction-following capabilities of DeepSeek-V3-0324. By interpolating their weights in various ways we construct a series of intermediate models, and found to our surprise that none of them seems to be defective. We specifically find that general reasoning performance tends to improve smoothly as R1's contribution increases. However, certain behavioral traits, such as the structured <think> reasoning traces characteristic of R1, emerge abruptly at a specific threshold in the merge ratio.

Geometrically, the merged models can be interpreted as residing in a low dimensional convex hull within the 671B-dimensional parameter space. Benchmarking models within this region reveals multiple configurations that outperform weaker parents and, in some cases, approach or exceed the stronger ones. These findings support the hypothesis that DeepSeek-V3, V3-0324, R1, and related fine-tunes occupy a shared loss valley, where interpolated models retain or recombine desirable traits without escaping into degraded regions of the landscape.

Motivated by our experiments and the modular structure of the MoE architecture, we explore merging the routed expert tensors of V3-0324 and R1. This results in *DeepSeek-R1T-Chimera*, a model that retains R1's reasoning performance while producing 40% fewer output tokens - a substantial reduction in verbosity with little to no loss in intelligence.

In addition to the promising benchmark results, R1T-Chimera has proven effective in a range of applications. It can be used as a replacement for R1, and we have deployed it for internal use on our GPU cluster. Based on positive feedback, we open-sourced it on Hugging Face [11]. As of late May 2025, it processed close to 5 billion tokens per day via the Chutes serverless AI platform [12].

These results highlight the promise of assembling structural components of MoEs as an efficient method for constructing high-performance LLMs.

2 Related Work

2.1 Efficient Reasoning

Reasoning models such as OpenAI's o1 [13], DeepSeek-R1 [14], Google's Gemini 2.5 [15] series, and Alibaba's Qwen3 [16] use a step-by-step "thinking process" to generate responses. This Chain-of-Thought (CoT) reasoning helps solve complex problems in mathematics, logic and programming, but often leads to verbose answers and consequently higher inference cost [17].

Efficient reasoning aims to preserve the performance while reducing the length and redundancy of these reasoning traces. This is desirable since reducing the inference costs makes models more economically viable, while faster response times improve user experience, and lower computational demand supports environmental sustainability.

Three main approaches to efficient reasoning were identified by [18]:

- *Input-based methods* optimize the prompt, for example by explicitly instructing the LLM to limit the length of its reasoning output [19]: "Let's think step by step and use less than `$BUDGET` tokens".
- *Output-based methods* reduce how much text is generated by the thinking model, for example by using a smaller model to suggest parts of an answer [20].
- *Model-based methods* modify the weights of the model, for example by fine-tuning reasoning models with a loss function that rewards shorter but still correct CoT reasoning [21, 19, 22, 23].

A more recent model-based approach is to merge models [24, 10]. For instance, [24] compared two approaches: fine-tuning a model with a loss function that penalizes long answers vs. merging the weights of two models—one optimized for short CoT reasoning and the other for long CoT. They found that fine-tuning with a length-penalizing loss function led to more token-efficient models than the model-merging method. We defer to [18] for a detailed overview of input and output-based efficient reasoning approaches.

2.2 Model Merging

Model merging, sometimes colloquially called *model souping*, refers to the process of combining multiple models into a single new model. Unlike *transfer learning*, which involves gradient-based fine-tuning on task-specific datasets and often requires significant computing resources, model merging does not require additional training. It has been applied effectively to small models in order to improve visual understanding [1], Japanese language capabilities [8], problem-solving abilities [3] and more recently, for efficient reasoning [10] or the curation of mixed datasets [25].

Some relevant forms of model merging are:

- *Average merging* [1]: This means averaging the weights of two models; at their time, [1] achieved state-of-the-art top-1 accuracy on ImageNet by merging two fine-tuned vision transformers.
- *Task arithmetic* [26]: This adds differences between parameter values of fine-tuned models and the original model, called "task vectors", to that of the original model. Mathematically, this method generalizes average merging.
- *TIES* [2]: Like task arithmetic, TIES-merging is a task-vector based method. It consists of the following 3 steps: (1) *T*R*I*M the task vectors to consider only significant differences of parameters values; (2) *E*LECT the *S*IGN for the change of each parameter via majority voting of different tasks or via alignment; and (3) MERGE the models by weighted aggregation of significant parameter changes. Applying TIES-merging to the dense DeepSeek models, DeepSeek-Math-7B-RL [27] with DeepSeek-Coder-Instruct-v1.5 [28], resulted in a model that outperformed both parents on reasoning benchmarks [29].

For a more comprehensive overview of merging techniques, see [10, 30].

While many great successes have been achieved by several merging techniques, challenges remain. For instance, [10] report that merging large-scale models (14B and 32B parameters) made it difficult to preserve reasoning quality when attempting to significantly shorten responses.

Our method might be best considered to build upon average merging, task arithmetic and some aspects of TIES merging. While our merging approach is more general, in R1T-Chimera, we optimized the trade-off between reasoning capability and inference cost by integrating the reasoning capabilities of DeepSeek's R1 671B with the more concise answering style of DeepSeek's V3-0324 671B. R1T-Chimera is, to the best of our knowledge, the first model merge to improve very large models.

2.3 Mixture of Experts

Mixture-of-Experts (MoE) architectures like Mistral's Mixtral-8x7B [31], DeepSeek-V3 [32], and Qwen's Qwen3 [16] divide a transformer's feedforward layers into multiple "experts", routing each token to only a subset of these experts [33]. This architecture has achieved state-of-the-art results with exceptional efficiency [34, 31]. For example, Mistral released Mixtral-8x7B [31] in late 2023, featuring 46.7 billion total parameters,

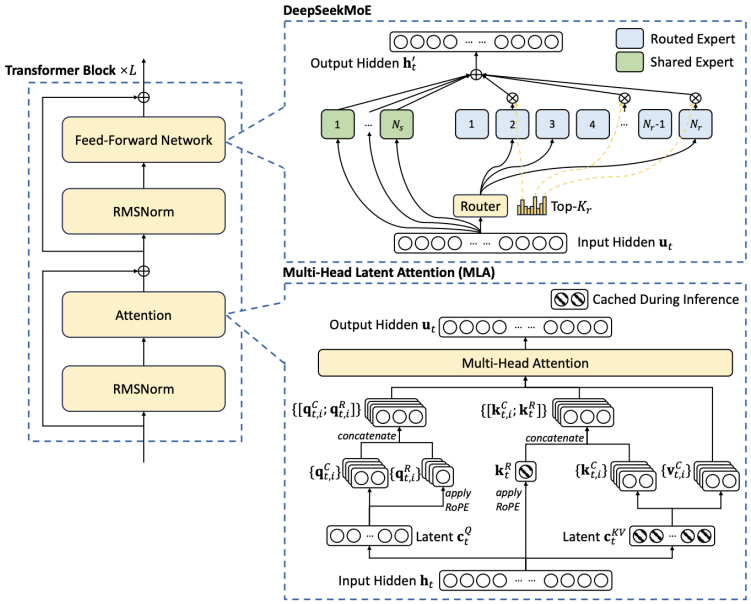


Figure 1: Basic architecture of DeepSeek-V3. (Illustration taken from [32], p.7.)

with 13 billion active during inference. Despite its significantly smaller active parameter count, Mixtral-8x7B matched the performance of the much larger LLaMA-2-70B model while being six times faster.

DeepSeek has made several key contributions to MoE architectures. Its May-2024 release, DeepSeek-V2 [35], was the strongest sparsely activated open-weights MoE model at the time. Since then, up to the time of publishing this article, DeepSeek has released six models of the DeepSeek-V3-MoE architecture. This architecture is a decoder-only transformer with a total of 671 billion parameters, but only 37 billion parameters are active per token during inference—approximately 1/18th of the model. This sparse activation makes it considerably faster than dense models of a similar size. The architecture, as depicted in Figure 1, combines both *shared experts*, which process every token to capture general patterns and ensure consistency across different inputs, with *routed experts*—selected dynamically per token via a trained router. The router activates 8 out of 256 experts. At its core, the model consists of a sequence of decoder blocks with multi-head self-attention, allowing it to model contextual relationships.

The following model variants got released by DeepSeek:

- DeepSeek-V3, released in December 2024, is a strong open-weights MoE model [32]. This model served as a foundation for the following fine-tuned specialist models.
- DeepSeek-R1 and DeepSeek-R1-Zero, released in January 2025, are two reasoning models, i.e., models that produce CoT at inference time [14]. Their CoT reasoning is placed between two specific tokens: <think> and </think>. In particular, R1 gained attention as the first reasoning model to fully disclose its CoT and for its innovative training approach via reinforcement learning from human feedback.
- Next, DeepSeek-V3-0324 was released in March 2025 [36]. This is an improved version of V3 with broader skills, specifically regarding coding, structured output and function calling/tool use.
- DeepSeek-Prover-V2-671B was released in May 2025, a model specifically trained for formal theorem proving using the programming language Lean 4 ([37, 38]).
- Shortly before we published this work, DeepSeek released DeepSeek-R1-0528 in May 2025 ([39]), an update of R1. While not yet included in the experiments of this publication, we are already evaluating its capabilities.

3 Methodology

This section describes how large language models like DeepSeek-R1T-Chimera can be derived by merging models.

We consider a set of models $M^{(i)}$, $i = 1, ..., n$ which share the same architecture. This setting arises, for instance, when each $M^{(i)}$, $i = 2, ..., n$ is obtained by fine-tuning a pre-trained model $M^{(1)}$. The common architecture implies the existence of corresponding weight tensors $W_l^{(i)}$ for each tensor index $l \in \mathcal{L}$. We construct a merged model, denoted $M^{(*)}$, whose tensors $W_l^{(*)}$ are derived from the original tensors $(W_l^{(i)})_{i,l}$ as follows:

- We select a subset $\mathcal{S} \subset \mathcal{L}$ of tensors that are to be merged. This subset may include all tensors or, for instance, only routed experts (cf. Section 4). The remaining tensors are taken from the base model $M^{(1)}$.
- Each model is weighted by a coefficient $\lambda_i, i = 1, ..., n$. In most cases, we consider convex combinations of tensors, i.e., requiring $\lambda_i \geq 0$ and $\sum_{i=1}^{n} \lambda_i = 1$. More generally, one can assign weights individually by tensor (and then obtain weights $\lambda_{i,l}$ for each model and tensor), but we stick to a fixed coefficient per model for simplicity, here.
- Inspired by the parameter-"trimming" approach in [2], we restrict the tensors that are to be merged to those which considerably differ between the models. Technically, we define a threshold $\delta \geq 0$ and include only those for which the maximum of the normalized Frobenius norms of the tensor differences (i.e., the Frobenius norms divided by the square root of the number of elements in the tensors) between the chosen base model $M^{(1)}$ and the other models $M^{(i)}$ $(i = 2, ..., n)$ exceeds δ.

Formally, this amounts to the following definition of the tensors $W_l^{(*)}$ of the merged model $M^{(*)}$:

$$W_l^{(*)} := \begin{cases} \sum_{i=1}^{n} \lambda_i W_l^{(i)} & \text{if } l \in \mathcal{S} \text{ and } \max_{i=2,\dots,n} \left\| W_l^{(1)} - W_l^{(i)} \right\|_{\text{F, norm.}} > \delta \\ W_l^{(1)} & \text{otherwise} \end{cases} \qquad \forall l \in \mathcal{L}. \tag{1}$$

3.1 Scenarios

We focus on a specific example with $n = 2$, DeepSeek-V3-0324 as the base model $M^{(1)}$ and DeepSeek-R1 as $M^{(2)}$. To explore the behavior of the merged model under different configurations, three scenarios are defined according to how the parameters in (1) are set.

3.1.1 Weighted-Average Merging

The relative contributions of V3-0324 and R1 to the merged model are controlled by weighting coefficients. Specifically, the merged weights are computed as a convex combination, with $\lambda_1 \in [0, 1]$ denoting the weight assigned to V3-0324 and $\lambda_2 := 1 - \lambda_1$ to R1.

This includes the special case $\lambda_1 = \lambda_2 = 0.5$, which corresponds to uniform averaging, as in standard model merging [1]. At the extremes, $\lambda = (0, 1)$ results in all merged tensors $\mathcal{S}$ being taken from R1, while $\lambda = (1, 0)$ simply yields the original V3-0324 base model.

3.1.2 Thresholding

Thresholding is used to control which tensors are included in the merge, based on their difference from the base model. A tensor is merged only if the normalized Frobenius norm of its difference with the base model exceeds a threshold $\delta > 0$ (cf. Section 3, eq. (1)).

While we do not fine-granularly trim on the level of single parameters but on the level of whole tensors, this thresholding still aims to focus the merge on relevant differences between the base and the other models. Our original intention here follows the ideas of [2], who apply trimming to avoid undesirable effects that stem from redundant adaptions in the different models compared to their common ancestor.

In order to approach thresholding more systematically, let us consider the actual tensor differences between V3-0324 and R1. Figure 2 illustrates the normalized Frobenius norms of the differences of the tensor groups of interest between V3-0324 and R1, i.e., $\left\|W_l^{(1)} - W_l^{(2)}\right\|_{\text{F, norm.}}$. Furthermore, Figure 3 shows the resulting distribution of normalized Frobenius differences of the tensors.

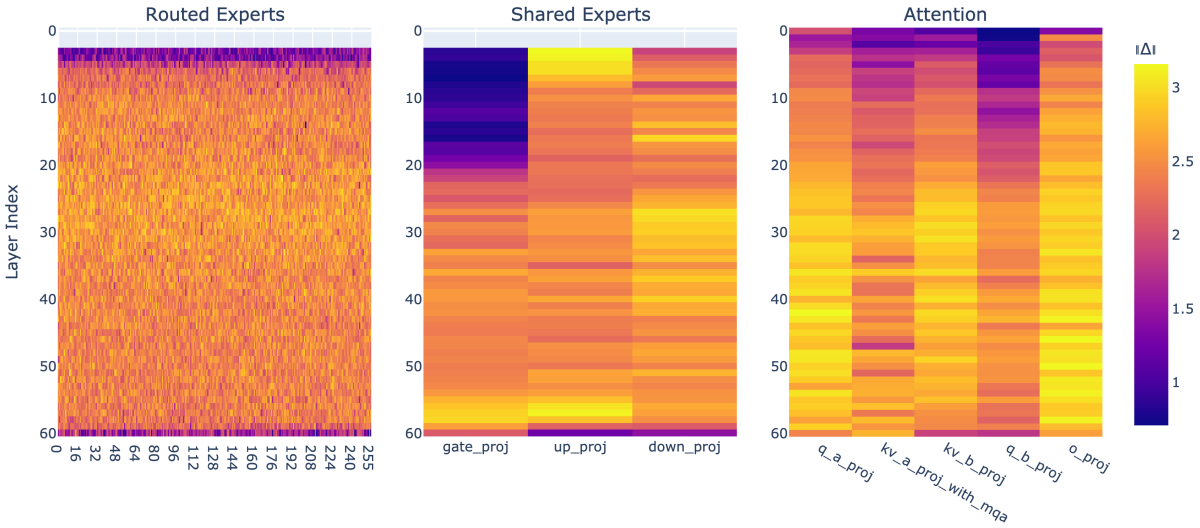


Figure 2: Illustration of the tensor differences between DeepSeek-R1 and DeepSeek-V3-0324 categorized into routed experts MLP, shared experts MLP and all attention blocks. The vertical axis represents the 61 decoder layers, where the first layer has index 0 and only layers from index 3 onward include experts. Notably, structural differences can be observed both w.r.t. the layers as well as the tensor category. Expert tensors differ rather regularly throughout the model, with clear patterns in the first and final layers. Shared experts even more clearly show a pattern with the gate-projection tensors being almost the same in the first but greatly different in the final layers. The attention tensors reveal similar effects.

3.1.3 Expert vs. Full Merging

The fine-grained experts substructure of tensor objects in a sparse MoE architecture allows for a multitude of merging strategies to derive new model instances. Our experiments are therefore just the very beginning of an exploration of the DeepSeek-MoE-model landscape.

There are several groups of tensors that may or may not be considered for merging: One may distinguish between tensors of the first *3 non-expert* decoder layers and the remaining *58 expert* decoder layers, between tensors corresponding to the *attention* blocks and tensors belonging to the subsequent (classic or expert) MLP blocks. Also, with regard to expert blocks, one may discriminate between *shared* and *routed experts* as well as the *gates* that determine which routed experts get activated during each forward path (cf. Figure 1).

For our experiments, we chose two subsets of tensors:

- *Full-merging subset*: All tensors are included in the merge, i.e, $\mathcal{S} := \mathcal{L}$.
- *Expert-merging subset*: Only tensors from the routed-experts blocks are merged, excluding the gating tensors.

3.2 Implementation of the Merge Method

The merging method expressed by eq. (1) is implemented using PyTorch [40]. The 671 billion model parameters of the DeepSeek-V3 model family are directly accessible by parsing the models' weights files, stored in

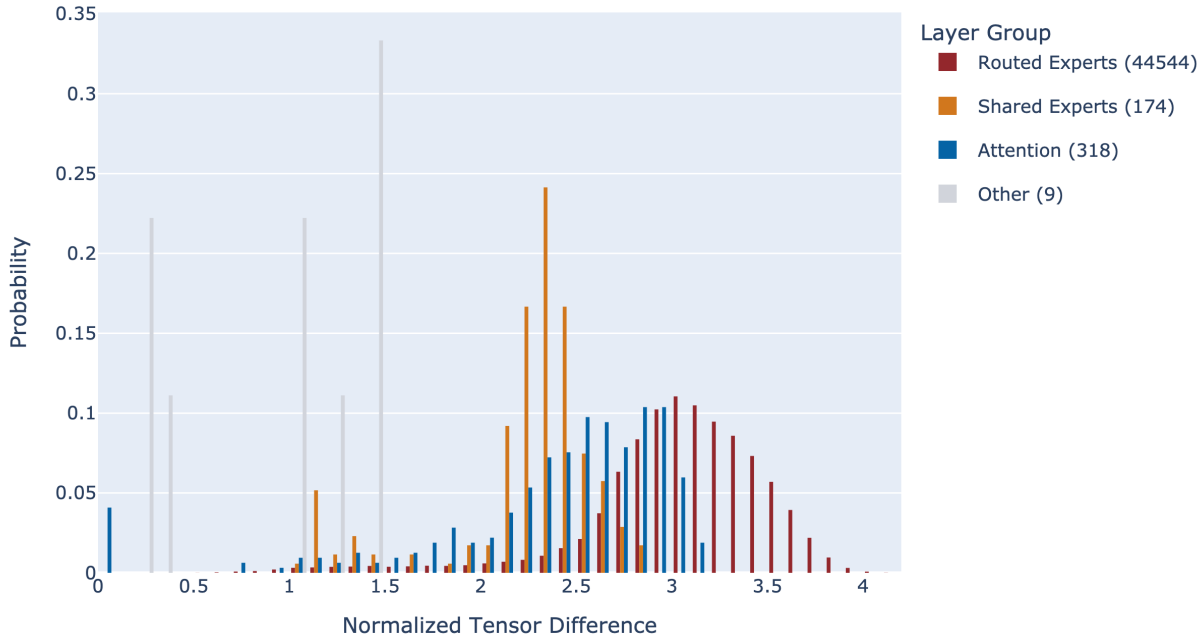


Figure 3: Distribution of normalized tensor differences, restricted to tensors with a normalized difference $\geq 10^{-3}$. It shows that the distributions of the tensor groups of interest differ significantly.

the `.safetensors` format. The model merging procedure can be performed by sequentially iterating over the 91,991 tensor objects that are in total contained in those files. After saving tensors of the merged model in the same structure and format as that of the source models, the merged model can be readily served by referencing the newly generated `.safetensors` files.

3.3 Experimental Setup

Models were evaluated on two hardware configurations: an 8×NVIDIA H100 94GB NVL cluster and an 8xAMD MI325X 256GB cluster capable of deploying several model variants simultaneously. Inference was carried out using a standard version of the vLLM inference engine [41] without modifications.

In order to validate the quality of a merged model, we benchmarked each instance primarily with MT-Bench [42] and AIME-2024 [43]. To classify a model as reasoning or non-reasoning, we enforced each model answer to start with the <think> token and measured the frequency of answers containing a closing </think> token. The occurrence of </think> indicates that the model has switched from Chain-of-Thought reasoning to providing the final answer. We will relate to the corresponding metric as </think>-tag/reasoning frequency (cf. Section 4.2). When answering MT-Bench or AIME-24 questions, the R1 and R1-Zero model have a reasoning frequency of 1, while V3 and V3-0324 have a reasoning frequency of 0.

4 Results

4.1 Reasoning Efficiency

Across the whole parameter space that we searched, the Assembly of Experts method successfully generated functional models that maintain characteristics of their parents to varying degrees. Notably, no parameter sets yielded poorly performing or broken models, hinting at a general robustness of the method.

As a proxy for general reasoning ability, we define an "intelligence score" as the average result over the AIME-2024 benchmark and the MT-Bench questions. Figure 4 shows this intelligence score plotted against the average output-token count as a proxy for intelligence cost. Several variants approach R1-level performance while generating substantially fewer tokens, suggesting that the reasoning efficiency improved.

4.2 On the Path to R1: Weight Analysis

Increasing the R1 fraction in merged models leads to a sudden rise in inference cost when approaching an equal weighting of the two models (i.e., $\lambda_2 \approx 0.5$). Figure 5 shows inference cost, measured via output-token count, as a function of the R1 contribution λ_2. This increase is less dominant when only merging the R1 experts compared to merging the full model.

We also examined a behavioral signal in the form of the </think>tag appearing in responses. V3-0324 does not produce these tags, while R1 was trained to produce them as part of its reasoning traces. Figure 6 shows a sharp transition around an R1 fraction of 0.5: merged models with an R1 contribution of 0.504 or higher usually emit the tag, whereas those with a greater V3-0324 proportion generally do not.

4.3 Thresholding Merges

As introduced in Section 3.1.2, we explored the effect of applying a non-zero threshold δ in eq. (1) to merge only those tensors where the divergence between V3-0324 and R1 exceeds δ. Figure 7 shows how varying this threshold affects the intelligence and inference cost of a model with equal weighting of V3-0324 and R1, (i.e., $\lambda_1 = \lambda_2 = 1/2$).

Our choices of threshold values were informed by the distribution of normalized tensor differences shown in Figure 3. At $\delta \approx 1.5$, the majority of attention tensors and routed-experts tensors are included in the merge. Increasing the threshold to $\delta = 2.5$ leads to the exclusion of most shared-expert tensors, effectively retaining their V3-0324 parameters. For $\delta > 2.5$, attention tensors are also excluded, and only the most divergent routed-expert tensors continue to be merged.

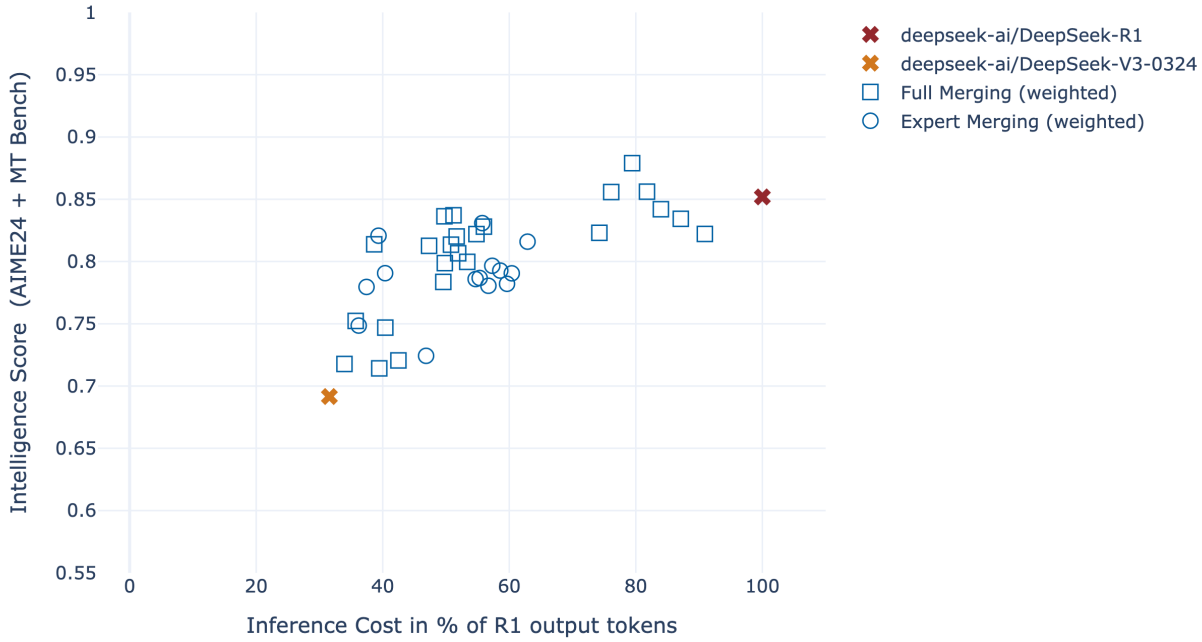


Figure 4: Intelligence score against the resulting inference cost by means of the average DeepSeek-R1 output-token count.

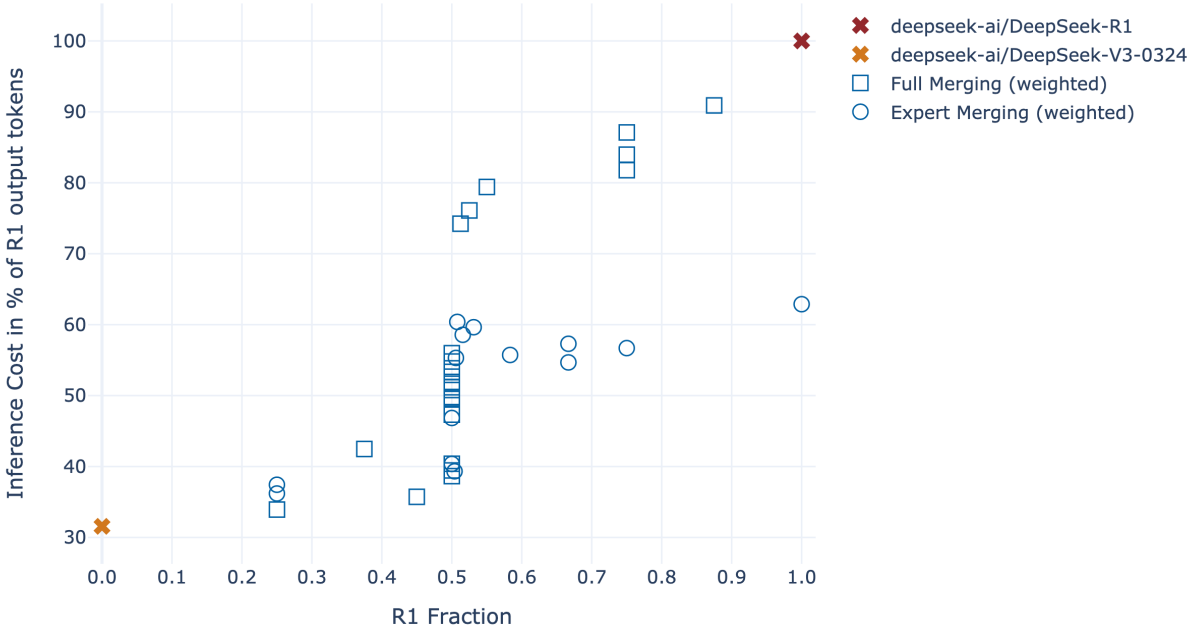


Figure 5: Relative inference cost measured in the number of output tokens relative to the ones produced by DeepSeek-R1 in our experiments. Squares depict models resulting from weighted-average merges of V3-0324 and R1, while circles depict weighted-average merges but restricted to the routed-expert tensors. In particular, we observe that merging only routed-expert tensors drastically reduces inference costs. Still, in both scenarios there is a steep sigmoid-like increase around an equal mix at 0.5.

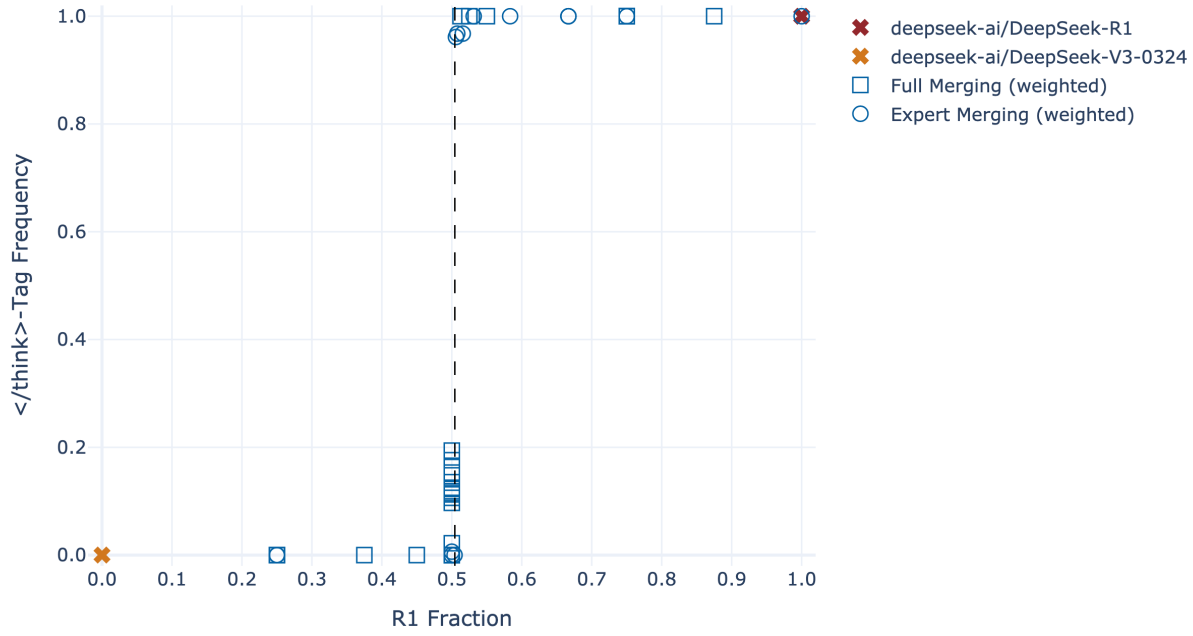
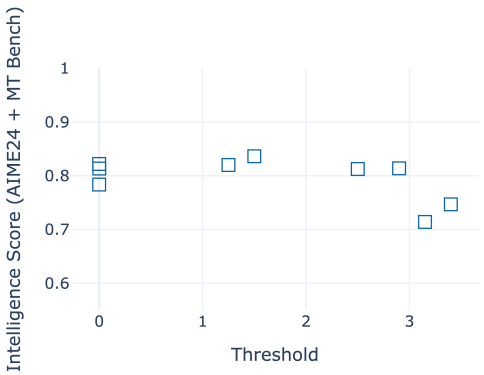
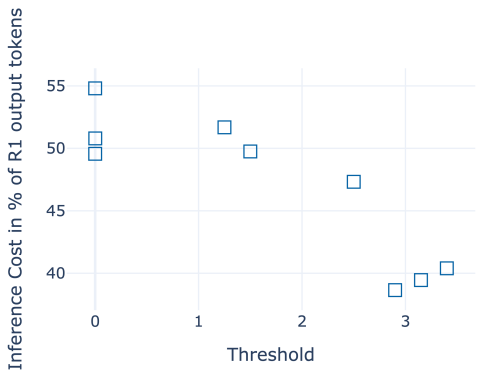


Figure 6: Frequency of a </think>tag in the model response, which can be regarded a proxy for the behavioral change to a reasoning model on the path from V3-0324 to R1. We noticed an almost clear cut around a R1 fraction of 0.504. Compared to Figure 5, this also holds true for expert-merging models despite their reduced token output.



(a) Intelligence scores for varying threshold values δ.



(b) Inference costs for varying threshold values δ.

Figure 7: Results for several threshold values δ in eq. (1) for the case of full merging with equal V3-0324 and R1 ratio.

As shown in Figure 7a, model performance remains stable up to a threshold of approximately $\delta = 3$. Beyond this point, as fewer R1 routed experts are used in the merge, the intelligence score begins to decline. Meanwhile, Figure 7b shows a consistent reduction in inference cost with increasing δ.

4.4 The R1T Chimera Model

The observations of Section 4.3 in combination with those of Section 4.2 motivate the following hypotheses:

- **Hypothesis 1:** The routed-experts tensors in DeepSeek-R1 play a central role in its enhanced reasoning capabilities.
- **Hypothesis 2:** The remaining tensors from V3-0324 (e.g., of attention layers, shared experts) are sufficient to coordinate or interface with these reasoning components without causing substantial degradation in performance.

One may thus explicitly consider the extreme case of using only the expert-merging subset with an extreme distribution of $\lambda = (0, 1)$, i.e., taking the full R1 routed experts.

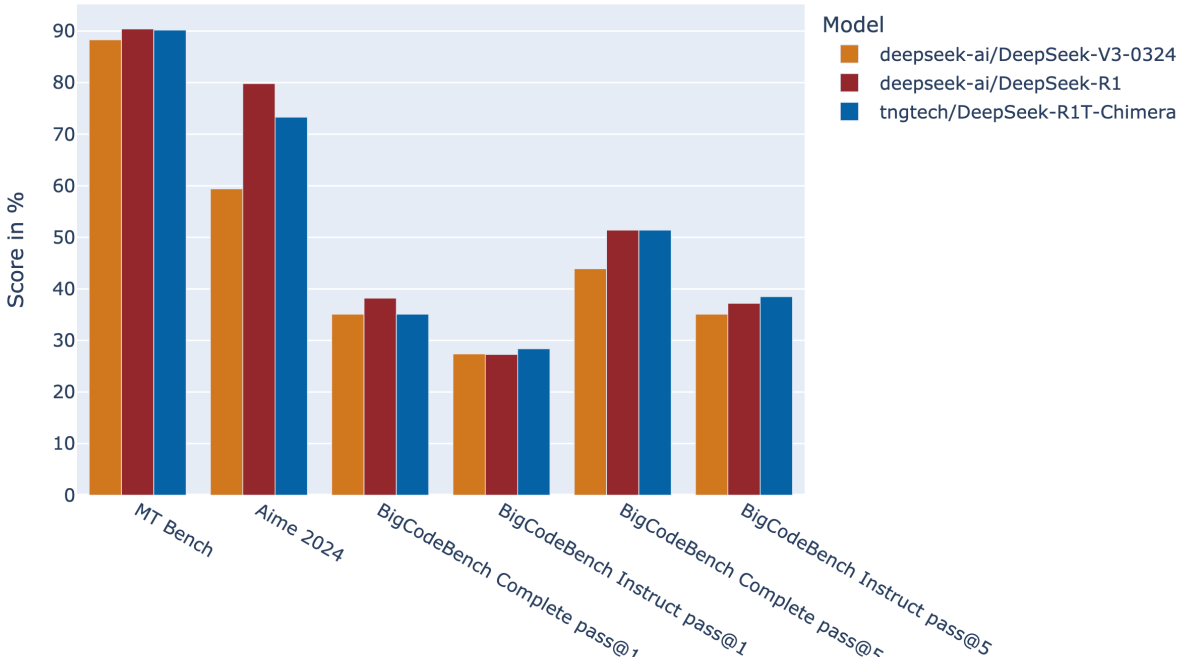


Figure 8: Benchmark results for DeepSeek-R1T-Chimera.

Figure 8 shows benchmark results for the resulting model, which we refer to as *DeepSeek-R1T-Chimera*. In addition to MT-Bench and AIME-2024, we evaluated the model on BigCodeBench ([44]), another well-known coding benchmark that assesses the capabilities for code generation and instruction following. While the set of benchmarks is not fully comprehensive, the results indicate that DeepSeek-R1T-Chimera performs competitively across a diverse range of tasks.

Following further internal validation, we released DeepSeek-R1T-Chimera on Hugging Face [11].

5 Conclusion

We merged very large language models based on the DeepSeek-V3-MoE architecture, focusing on combining DeepSeek V3-0324 with R1. To the best of our knowledge, this is the first successful attempt to merge models of this scale, and we demonstrated its effectiveness by combining the answering style of V3-0324 with the reasoning capabilities of R1, resulting in the model DeepSeek-R1T-Chimera.

By varying the proportion of weights taken from R1, we explored a continuous interpolation space. Surprisingly, all merges resulted in working and capable variants. While some of their properties like their general intelligence changed gradually in this space, some behavioral traits emerged with a sharp transition near equal weighting, where the merged models began producing reasoning output using the <think>...</think>-construct of R1. Our findings suggest that the parameter space between the specialized fine-tuned models V3-0324 and R1 does not solely contain inferior models. Instead, we have identified several sweet spots along paths through the valley in the loss landscape.

In particular, recalling the MoE architecture with the specialized experts separated from the remaining parts of the network motivated us to merge those two network parts individually. Finally, this resulted in the DeepSeek-R1T-Chimera model. This model demonstrates improved reasoning efficiency while maintaining strong performance.

We believe that this is just the beginning - this technique can be applied to future fine-tuned variants of the DeepSeek-V3-MoE architecture to build more efficient reasoning models, and more generally, to combine other desirable traits.

Acknowledgements

谢谢, vielen Dank, and thanks to DeepSeek for their innovative work and for making the V3 and R1 models available. Without these models, this research would not have been possible. We also thank the open-source community, particularly the teams behind vLLM and Hugging Face, and the many contributors whose tools were used in our experiments. Also thanks and kudos to Cohere for their great Command A paper which appeared while we were working on the Chimera. Special thanks go to OpenRouter and Chutes for making our R1T Chimera model freely available on their large infrastructure. Danke an Lennard Schiefelbein from TNG for his swift support in running our benchmark evaluations. Last not least: Thanks to everyone who ever made model merging work in an experiment, wondered why it worked, couldn't quite explain it - but still was daring enough to publish it.

References

[1] Mitchell Wortsman et al. "Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. PMLR. 2022, pp. 23965–23998.

[2] Prateek Yadav et al. "TIES-merging: Resolving interference when merging models". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 7093–7115.

[3] Ken Shoemake. "Animating rotation with quaternion curves". In: *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*. 1985, pp. 245–254.

[4] MohammadReza Davari and Eugene Belilovsky. "Model breadcrumbs: Scaling multi-task model merging with sparse masks". In: *European Conference on Computer Vision*. Springer. 2024, pp. 270–287.

[5] Le Yu et al. "Language models are Super Mario: Absorbing abilities from homologous models as a free lunch". In: *41st International Conference on Machine Learning*. 2024.

[6] Sidak P Singh and Martin Jaggi. "Model fusion via optimal transport". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 22045–22055.

[7] Moritz Imfeld et al. *Transformer fusion with optimal transport*. 2023. arXiv: 2310.05719.

[8] Takuya Akiba et al. "Evolutionary optimization of model merging recipes". In: *Nature Machine Intelligence* (2025), pp. 1–10.

[9] Team Cohere. *Command A: An Enterprise-Ready Large Language Model*. 2025. arXiv: 2504.00698 [cs.CL]. URL: https://arxiv.org/abs/2504.00698.

[10] Han Wu et al. *Unlocking efficient long-to-short LLM reasoning with model merging*. 2025. arXiv: 2503.20641.

[11] TNG Technology Consulting GmbH. *DeepSeek R1T Chimera*. https://huggingface.co/tngtech/DeepSeek-R1T-Chimera. Hugging Face Model Card. Apr. 2025.

[12] Rayon Labs. *Chutes*. URL: https://chutes.ai (visited on 05/27/2025).

[13] OpenAI. *Learning to reason with LLMs*. URL: https://openai.com/index/learning-to-reason-with-llms/ (visited on 05/27/2025).

[14] DeepSeek-AI. *DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning*. 2025. arXiv: 2501.12948.

[15] Koray Kavukcuoglu. *Gemini 2.5: Our most intelligent AI model*. URL: https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#gemini-2-5-thinking (visited on 05/27/2025).

[16] Qwen Team. *Qwen3 technical report*. 2025. arXiv: 2505.09388.

[17] Xingyu Chen et al. *Do NOT think that much for 2+ 3=? On the overthinking of o1-like LLMs*. 2024. arXiv: 2412.21187.

[18] Yang Sui et al. *Stop overthinking: A survey on efficient reasoning for large language models*. 2025. arXiv: 2503.16419.

[19] Tingxu Han et al. *Token-budget-aware LLM reasoning*. 2024. arXiv: 2412.18547.

[20] Baohao Liao et al. *Reward-guided speculative decoding for efficient LLM reasoning*. 2025. arXiv: 2501.19324.

[21] Daman Arora and Andrea Zanette. *Training language models to reason efficiently*. 2025. arXiv: 2502.04463.

[22] Yu Kang et al. "C3oT: Generating shorter Chain-of-Thought without compromising effectiveness". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. 23. 2025, pp. 24312–24320.

[23] Tengxiao Liu et al. *Can language models learn to skip steps?* 2024. arXiv: 2411.01855.

[24] Kimi Team. *Kimi k1.5: Scaling reinforcement learning with LLMs*. 2025. arXiv: 2501.12599.

[25] Zhixu Silvia Tao et al. *Merge to mix: Mixing datasets via model merging*. 2025. arXiv: 2505.16066.

[26] Gabriel Ilharco et al. *Editing models with task arithmetic*. 2022. arXiv: 2212.04089.

[27] Zhihong Shao et al. *DeepSeekMath: Pushing the limits of mathematical reasoning in open language models*. 2024. arXiv: 2402.03300.

[28] Daya Guo et al. *DeepSeek-Coder: When the large language model meets programming–the rise of code intelligence*. 2024. arXiv: 2401.14196.

[29] Yiming Zhang et al. *Unconstrained model merging for enhanced LLM reasoning*. 2024. arXiv: 2410.13699.

[30] Weishi Li et al. *Deep model fusion: A survey*. 2023. arXiv: 2309.15698.

[31] Albert Q Jiang et al. *Mixtral of experts*. 2024. arXiv: 2401.04088.

[32] DeepSeek-AI. *DeepSeek-V3 technical report*. 2024. arXiv: 2412.19437.

[33] Robert A Jacobs et al. "Adaptive mixtures of local experts". In: *Neural computation* 3.1 (1991), pp. 79–87.

[34] Damai Dai et al. *DeepSeekMoE: Towards ultimate expert specialization in Mixture-of-Experts language models*. 2024. arXiv: 2401.06066.

[35] DeepSeek-AI. *DeepSeek-V2: A strong, economical, and efficient Mixture-of-Experts language model*. 2024. arXiv: 2405.04434.

[36] DeepSeek-AI. *DeepSeek-V3-0324*. https://huggingface.co/deepseek-ai/DeepSeek-V3-0324. Hugging Face Model Card. (Visited on 05/27/2025).

[37] ZZ Ren et al. *DeepSeek-Prover-V2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition*. 2025. arXiv: 2504.21801.

[38] Leonardo de Moura and Sebastian Ullrich. "The Lean 4 theorem prover and programming language". In: *Automated Deduction–CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings 28*. Springer. 2021, pp. 625–635.

[39] DeepSeek-AI. *deepseek-ai/DeepSeek-R1-0528*. https://huggingface.co/deepseek-ai/DeepSeek-R1-0528. Hugging Face Model Card. (Visited on 05/29/2025).

[40] Jason Ansel et al. "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation". In: *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. ASPLOS '24. La Jolla, CA, USA: Association for Computing Machinery, 2024, pp. 929–947. ISBN: 9798400703850. DOI: 10.1145/3620665.3640366. URL: https://doi.org/10.1145/3620665.3640366.

[41] Woosuk Kwon et al. "Efficient Memory Management for Large Language Model Serving with PagedAttention". In: *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*. 2023.

[42] Lianmin Zheng et al. "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 46595–46623.

[43] Mathematical Association of America (MAA). *American Invitational Mathematics Examination - AIME 2024*.

[44] Terry Yue Zhuo et al. *BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions*. 2025. arXiv: 2406.15877.