Revisiting Model Interpolation for Efficient Reasoning

Taiqiang Wu[◇] **Runming Yang**[◇] **Tao Liu**[♠] **Jiahao Wang**[◇] **Ngai Wong**[◇]

[◇]The University of Hong Kong [♠]Tsinghua University

takiwu@connect.hku.hk nwong@eee.hku.hk

Abstract

Model merging, typically on Instruct and Thinking models, has shown remarkable performance for efficient reasoning. In this paper, we systematically revisit the simplest merging method that interpolates two weights directly. Particularly, we observe that model interpolation follows a three-stage evolutionary paradigm with distinct behaviors on the reasoning trajectory. These dynamics provide a principled guide for navigating the performance-cost trade-off. Empirical results demonstrate that a strategically interpolated model surprisingly surpasses sophisticated model merging baselines on both efficiency and effectiveness. We further validate our findings with extensive ablation studies on model layers, modules, and decoding strategies. Ultimately, this work demystifies model interpolation and offers a practical framework for crafting models with precisely targeted reasoning capabilities. Code is available at Github.

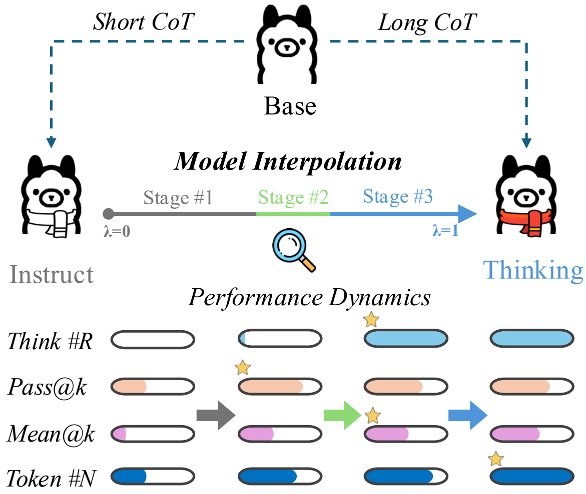

Figure 1: The performance dynamics for the model interpolation between Instruct and Thinking models. *Think #R* denotes the ratio of samples with *</think>* token in responses. *Token #N* denotes the number of tokens in responses.

1 Introduction

Large language models (LLMs), such as Qwen3 (Yang et al., 2025), OpenAI o1 (Jaech et al., 2024), and Deepseek R1 (Guo et al., 2025), have revolutionized the field of natural language processing (NLP). Their remarkable success in complex tasks is largely attributed to emergent reasoning capabilities, which benefit from scaled chain-of-thoughts (Wei et al., 2022) during test time (Snell et al., 2024). However, longer CoT also introduces significant trade-offs such as over-thinking (Chen et al., 2024) and high latency issues (Sui et al., 2025). Consequently, how to achieve efficient reasoning without compromising performance remains a critical challenge.

To address this challenge, model merging has emerged as a compelling solution (Yang et al., 2024; Wu et al., 2025c) The core idea is to merge the weights of two specialized models, including a Thinking model adept at long-CoT reasoning and an Instruct model optimized for short, direct answers, and thus create a hybrid model balancing reasoning capability with token efficiency (Team et al., 2025; Wu et al., 2025a; Ma et al., 2025). Existing merging methods can be categorized into weighted-based (Utans, 1996; Ilharco et al., 2022), subspace-based (Yadav et al., 2023; Yu et al., 2024), and routing-based (Muqeeth et al., 2023; Tang et al., 2024). Most of these merging methods require the paired pretrained models, a constraint not shared by the simpler method of direct model interpolation.

In this paper, we revisit the model interpolation (**MI**) method and systematically analyze the dynamics as the interpolation coefficient λ sweeps from 0 to 1. Surprisingly, we find that the performance metrics, such as Pass@k and Mean@k, do not evolve linearly but instead follow a distinct three-stage paradigm detailed in Figure 1. In stage #1, where the weights of the Instruct model are

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8624–8638
July 2-7, 2026 ©2026 Association for Computational Linguistics

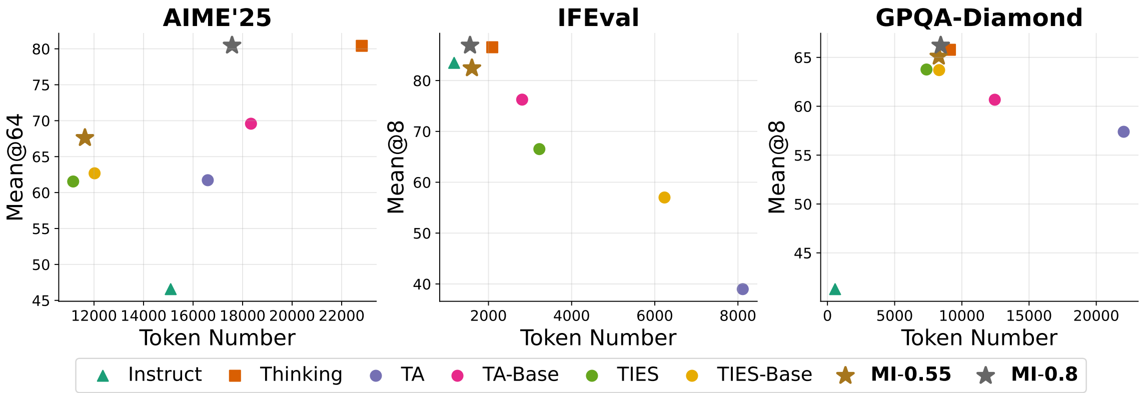

Figure 2: Performance of vanilla Instruct, Thinking, and model merging methods on AIME'25, IFEval, and GPQA-Diamond. **MI** denotes the model interpolation and the suffix for the interpolation coefficient λ. The results indicate that **MI** surpasses these baselines on both efficiency and effectiveness.

dominant, Pass@k and the number of output tokens increase rapidly with **almost no** explicit reasoning in responses. The thinking pattern, measured by the presence of token *</think>*, emerges in stage #2 with larger λ, with the Mean@k increasing faster than Pass@k. At stage #3, the output responses become substantially longer, yet the marginal gains in Pass@k and Mean@k diminish, corresponding to the overthinking phenomenon (Chen et al., 2024). These observations offer novel insights into the mechanics of weight interpolation and further provide a principled guide for desired reasoning behaviors.

We conduct extensive experiments on Qwen3 models (Yang et al., 2025), interpolating between the official Thinking and Instruct variants. As shown in Figure 2, a strategically interpolated model surpasses the mainstream model merging baselines across various challenging benchmarks, including mathematical reasoning (AIME'25 (AIME, 2025)), instruction-following (IFEval (Zhou et al., 2023)), and science reasoning (GPQA-Diamond (Rein et al., 2024)). Beyond this superior performance, our analysis provides a practical framework for crafting models with desired behaviors such as adhering to a specific token budget. We further conduct fine-grained ablations on model layers and modules, providing a comprehensive view of how interpolation fuses reasoning capabilities.

Our main contributions can be summarized as follows:

- We systematically revisit model interpolation methods and uncover a three-stage evolutionary paradigm. This framework provides principled guidance for efficient reasoning.
- We empirically demonstrate that a simple yet strategically interpolated model can surpass sophisticated merging baselines across a diverse suite of challenging benchmarks.
- We provide deep mechanistic insights through fine-grained ablation studies on layers, modules, and decoding strategies.

2 Related Work

2.1 Efficient Reasoning

Efficient reasoning aims to reduce the output tokens while preserving reasoning capability as much as possible (Sui et al., 2025). The methods can be categorized as 1) model-based, setting a short response as the optimization goal during SFT (Yu et al., 2025; Li et al., 2025) or RL (Team et al., 2025; Zhang et al., 2025a,b; Wu et al., 2026), 2) decoding-based, which modifies the output paradigm such as latent reasoning (Hao et al., 2024; Butt et al., 2025) and dynamic reasoning (Sun et al., 2024; Wang et al., 2025), and 3) prompts-based, refining the input prompts for enable concise and efficient reasoning (Xu et al., 2025; Aytes et al., 2025). We refer the reader to Sui et al. (2025) for a comprehensive survey. In this paper, we focus on the training-free model interpolation for efficient thinking.

2.2 Model Merging

Model merging methods merge the parameters of multiple separate models with different capabilities, and are widely applied for continual learning (Marczak et al., 2024), multi-task learning (Yang et al., 2023), and model attack (Gangwal and Sharma, 2025). The key is to merge the weights from different models following the same structure. One intuitive idea is to average the weights (Utans, 1996), while the task arithmetic framework extends the idea to the task vector (Ilharco et al., 2022). Please refer to Yang et al. (2024) for more details. Recently, Kimi k1.5 averages two models with long CoT and short CoT capabilities for efficient reasoning (Team et al., 2025), which can be viewed as a special case of model interpolation with coefficients λ being 0.5. In this work, we extend such idea and analyze the model interpolation with various coefficients.

3 Preliminary

3.1 Instruct and Thinking Models

Regarding the reasoning ability of LLMs, there are two distinct reasoning styles, i.e., long CoT and short CoT. Typically, long CoT (a.k.a, slow thinking) LLMs perform better on complex tasks with an explicit thinking process but incur greater inference latency, while short CoT (a.k.a, quick thinking) LLMs are optimized to produce short, direct answers, prioritizing speed and efficiency.

These two reasoning modes are typically realized through two primary paradigms. The first one is a hybrid reasoning supporting both Thinking and Non-thinking, such as `Qwen3-4B` (Yang et al., 2025). The second paradigm involves post-training two separate specialist models, such as `Qwen3-4B-Instruct-2507` for quick thinking and `Qwen3-4B-Thinking-2507` for slow thinking.

Table 1 reports the parameter similarity (Wu et al., 2025b) among Qwen3-4B series, revealing two key findings. First, all the paired models are highly similar in weights. Second, `Qwen3-4B` is more similar to `Qwen3-4B-Thinking-2507` models than `Qwen3-4B-Base`, suggesting a potential inheritance relationship. Also, `Qwen3-4B-Instruct` and `Qwen3-4B-Thinking` are more similar than other pairs, indicating that they **might be trained on the same prompts** with different responses.

Models		σ
`Qwen3-4B-Base`	`Qwen3-4B`	0.0326
`Qwen3-4B-Base`	`Qwen3-4B-Instruct`	0.0562
`Qwen3-4B-Base`	`Qwen3-4B-Thinking`	0.0638
`Qwen3-4B`	`Qwen3-4B-Instruct`	0.0589
`Qwen3-4B`	`Qwen3-4B-Thinking`	0.0633
`Qwen3-4B-Instruct`	`Qwen3-4B-Thinking`	0.0269

Table 1: Weight similarity σ (Wu et al., 2025b) on paired models from Qwen3 series. We omit the suffix `-2507` for simplicity. The smaller σ, the more similar.

3.2 Model Merging Methods

Given a set of T models, $\{\Theta^{(1)}, \dots, \Theta^{(T)}\}$, that share a common architecture and are typically trained based on the same pre-trained model $\Theta^{(0)}$, the objective of model merging is to produce a new model, $\Theta^{(\text{Merge})}$, through a parameter-wise operation:

$$\Theta^{(\text{Merge})} = \text{Merge}(\Theta^{(0)}, \Theta^{(1)}, \dots, \Theta^{(T)}). \quad (1)$$

An intuitive baseline strategy is to average the parameters of the models:

$$\Theta^{(\text{Merge})} = \frac{1}{T}\sum_{i=1}^{T} \Theta^{(i)}. \quad (2)$$

Task Arithmetic. Ilharco et al. (2022) defines task vector (TV) to represent the parameter shift on a specific task, which is calculated as:

$$TV^{(i)} := \Theta^{(i)} - \Theta^{(0)}. \quad (3)$$

Thus, multiple capabilities can be combined by aggregating their corresponding task vectors and adding them back to the base model:

$$\Theta^{(\text{Merge})} = \Theta^{(0)} + \alpha \sum_{i=1}^{T} TV^{(i)}, \quad (4)$$

where α is the scaling factor for all TVs.

TIES-Merging. TIES-Merging (Yadav et al., 2023) proposes to transform weights into sparse subspaces for merging due to the over-parameterized nature of neural networks (Choudhary et al., 2020). For a given task vector $TV^{(i)}$, TIES-Merging retains part of parameters with the highest magnitudes:

$$TV^{(i)}_{\text{TIES}} = \text{Top-k}(TV^{(i)}). \quad (5)$$

These sparse task vectors are then resolved by voting for signs and merged following Equation 4.

Model Interpolation (MI). In this paper, we focus on the task of efficient reasoning by merging a Thinking model $\Theta^{(\text{Thi})}$, and an Instruct model $\Theta^{(\text{Ins})}$. The model interpolation (MI) is formulated as:

$$\Theta^{(\text{Merge})} = \lambda\Theta^{(\text{Thi})} + (1 - \lambda)\Theta^{(\text{Ins})}. \quad (6)$$

This process is agnostic to the base model and can be framed as a special case of Task Arithmetic. By defining $TV^{(\text{Thi})} = \Theta^{(\text{Thi})} - \Theta^{(\text{Base})}$ and $TV^{(\text{Ins})} = \Theta^{(\text{Ins})} - \Theta^{(\text{Base})}$, it is easy to show that:

$$\begin{aligned} &\lambda\Theta^{(\text{Thi})} + (1-\lambda)\Theta^{(\text{Ins})} = \lambda(TV^{(\text{Thi})} + \Theta^{(\text{Base})}) \\ &+ (1-\lambda)(TV^{(\text{Ins})} + \Theta^{(\text{Base})}) \\ &= \Theta^{(\text{Base})} + \lambda TV^{(\text{Thi})} + (1-\lambda)TV^{(\text{Ins})}. \end{aligned} \tag{7}$$

The $\Theta^{(\text{Base})}$ can be an *arbitrary* model. This derivation shows that MI is equivalent to performing Task Arithmetic on the Thinking and Instruct task vectors with scaling factors of λ and $(1-\lambda)$, respectively.

4 Revisiting Model Interpolation

4.1 Experimental Setup

Models. We conduct experiments merging the Qwen3-4B and Qwen3-30B-A3B models. For both settings, we download the official weights from huggingface and merge the `Instruct-2507` and `Thinking-2507` variants.

Benchmarks. To ensure a comprehensive evaluation, we select three representative benchmarks that cover diverse reasoning skills, containing IFEval for instruction following (Zhou et al., 2023), GPQA-Diamond for scientific reasoning (Rein et al., 2024), and AIME'25 for mathematical reasoning (AIME, 2025). We adapt the OpenCompass (Contributors, 2023) framework for evaluation. Further details of these benchmarks are provided in Appendix A.

Decoding Strategy. For the baseline Instruct and Thinking models, we employ their official sampling configurations to ensure optimal performance. The Thinking model uses a temperature T of 0.6 and Top-p of 0.95, while the Instruct model uses 0.7 and 0.8, respectively. For all merged models, we consistently apply the same hyperparameters with Thinking model (i.e., $T = 0.6$, Top-p $= 0.95$), deferring a detailed analysis of hyperparameter sensitivity to Section 5.1. We roll out 64 times for AIME'25 and 8 for IFEval and GPQA-Diamond.

Evaluation Metrics. We evaluate the models across the following abilities:

- **Effectiveness.** We report Pass@k and Mean@k scores. For Pass@k, we adapt the unbiased estimator (Chen et al., 2021).
- **Consistency.** We also report Vote@k to measure the stability of the model's most frequent answer.
- **Efficiency.** We measure computational cost by the average number of tokens in the generated responses, denoted as Token #N.
- **Reasoning Pattern.** We further introduce the Thinking Ratio (Think #R), defined as the percentage of responses containing the *</think>* token, to quantify the prevalence of explicit CoT reasoning.

4.2 Three-Stage Paradigm

Figure 3 and Figure 6 illustrate the performance dynamics on the Qwen3-4B and Qwen3-30B-A3B models, respectively. Across both model scales, structures, and all three benchmarks, we observe that the performance dynamics do not evolve linearly with the interpolation coefficient λ. Instead, they follow a consistent and predictable three-stage paradigm, which we detail below using the Qwen3-4B model as the primary example.

Stage #1. Corresponding to $\lambda \in [0, 0.4)$ for Qwen3-4B model. In this initial phase, the merged model is dominated by the Instruct model but begins to incorporate traits from the Thinking model, and thus **generating longer outputs without adopting an explicit thinking process**.

The performance of different abilities distinct each other. The Think Ratio (*Think #R*) remains near zero. Hence, the model almost never generates explicit Chain-of-Thought steps, opting for direct answers. Meanwhile, the number of tokens (*Token #N*) and Pass@k gradually increase as the model begins to generate more verbose responses. For example, the *Token #N* on IFEval increases from 1174 to 6492. However, due to the lack of an explicit reasoning process, the Mean@k and Vote@k increase much more gently on AIME'25 and GPQA-Diamond. In addition, there is a *large drop* on IFEval, since some input questions require being answered with token limits.

Stage #2. Corresponding to $\lambda \in [0.4, 0.6]$ for Qwen3-4B. In this stage, the **reasoning pattern following Thinking models rapidly emerges**, leading to largely increased Mean@k and gently increased Pass@k and *Token #N*. This stage marks a critical and dramatic phase transition.

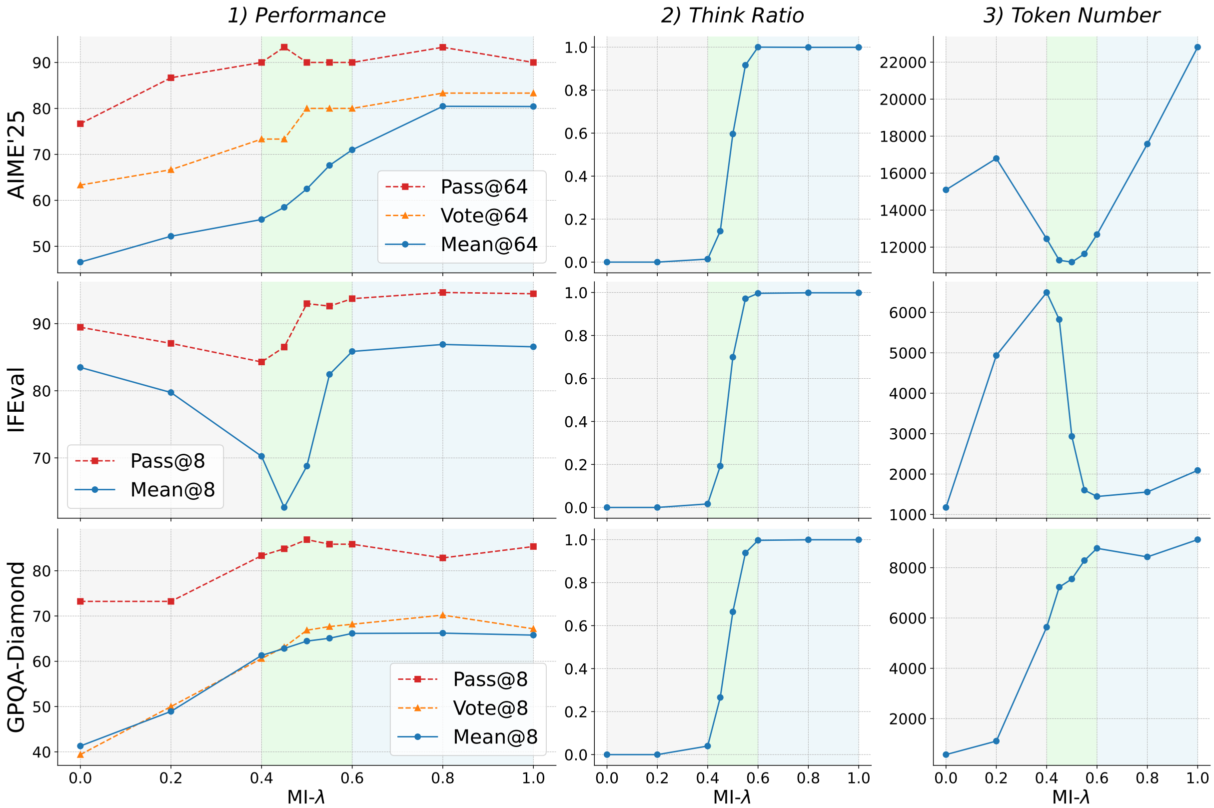

Figure 3: The performance dynamics of model interpolation (MI) on `Qwen3-4B-Instruct-2507` and `Qwen3-4B-Thinking-2507`. The dynamics follow a three-stage evolutionary paradigm colored in grey, green, and blue. λ denotes the interpolation coefficient ranging from 0 to 1. Please refer to Appendix B and Appendix C for results on `Qwen3-30B-A3B` and `Llama-3.1-8B`, respectively.

Specifically, the *Think #R* abruptly rises from nearly 0 to 1, indicating the rapid emergence of explicit long CoT capabilities from the Thinking model. Across the three benchmarks, all the metrics show gains in this stage. In contrast to stage #1, the Mean@k scores increase largely while the Pass@k scores more gently. For instance, the Mean@64 score increases from 55.8 to 71.0 on AIME'25, while Pass@64 is unchanged. The sharp increase in Mean@k suggests the dramatically improved quality of the reasoning process. Notably, the Pass@k score often reaches its peak at the end of this stage. This stage corresponds to a significant decrease in token count, indicating a potential sweet spot for effectiveness and efficiency.

Stage #3. Corresponding to $\lambda \in (0.6, 1.0]$ for Qwen3-4B. In this final stage, the merged model **converges** to the pure Thinking model, with **continuously increasing** *Token #N* and slight change in Pass@k and Mean@k.

At this stage, the *Think #R* is saturated at 1.0 and the *Token #N* continuously increases, reflecting the high cost of generating long-form reasoning for all inputs. Although Mean@k continues to show slight improvements, Pass@k often plateaus or even slightly declines from its peak at Stage #2. This suggests a point of diminishing returns and provides clear evidence of the over-thinking phenomenon (Chen et al., 2024), where longer reasoning does not necessarily lead to better performance. Interestingly, the merged model can outperform the pure Thinking model ($\lambda = 1.0$) at certain points in this stage (e.g., at $\lambda = 0.8$), demonstrating that a slight blend with the Instruct model can sometimes regularize the reasoning process and yield better performance (Wu et al., 2025c).

Discussion on Larger Model. The performance dynamics of the much larger Qwen3-30B-A3B models (shown in Figure 6) follow a similar three-stage paradigm, confirming the generalization of our findings. However, the specific ranges for each stage differ, with Stage #2 occurring later, at $\lambda \in [0.5, 0.8]$. Experiments on Llama-3.1-8B indicate a consistent conclusion across different stages. Further details are available in Appendix B and Appendix C.

Method	AIME'25			IFEval			GPQA-Diamond		
	Mean/Pass @64	*Token #N*	*Think #R*	*Mean/Pass @8*	*Token #N*	*Think #R*	*Mean/Vote @8*	*Token #N*	*Think #R*
Instruct	46.6/76.7	15097	0.0	83.5/89.5	1174	0.0	41.3/39.4	570	0.0
Thinking	80.4/90.0	22813	99.9	86.6/94.5	2091	100.0	65.8/67.2	9114	99.9
TA	61.7/86.7	16594	0.5	39.0/60.8	8116	7.6	57.4/59.6	22042	0.2
TA-Base	69.6/86.7	18339	43.0	76.3/91.9	2810	76.5	60.7/66.2	12450	32.9
TIES	61.5/93.3	11159	65.4	66.5/90.8	3224	71.2	63.8/68.2	7369	72.2
TIES-Base	62.7/90.0	12024	15.6	57.0/87.8	6234	21.6	63.7/66.7	8309	21.8
DARE	61.4/93.3	11231	38.0	62.8/91.7	2758	51.7	64.4/67.7	7215	58.4
SLERP	62.6/90.0	10944	60.4	68.7/92.1	3220	68.2	65.2/69.2	7523	65.2
MI-0.2	52.2/86.7	16794	0.0	79.7/87.1	4933	0.0	48.9/50.0	1107	0.0
MI-0.4	55.8/90.0	12448	1.4	70.2/84.3	6492	1.6	61.3/60.6	5634	4.0
MI-0.5	62.5/90.0	11189	59.6	68.8/93.0	2932	69.9	64.5/66.7	7548	66.4
MI-0.6	71.0/90.0	12681	100.0	85.9/93.7	1445	99.6	66.2/68.2	8769	99.7
MI-0.8	**80.5/93.3**	17574	99.9	**86.9/94.6**	1556	99.9	**66.2/70.2**	8427	99.9

Table 2: Performance comparison across AIME'25, IFEval, and GPQA-Diamond when merging `Qwen3-4B-Instruct-2507` and `Qwen3-4B-Thinking-2507`. We sample 64 times on AIME'25 and 8 times on others. Since questions from GPQA-Diamond are multiple-choice, we report Vote@8 instead of Pass@8.

4.3 Compared with More Baselines

We further compare MI against several model merging baselines, including Task Arithmetic (TA) (Ilharco et al., 2022) and TIES-Merging (TIES) (Yadav et al., 2023), which are detailed in Section 3.2. We further compare to two extra baselines, i.e., DARE (Yu et al., 2023) and SLERP (Goddard et al., 2024). One critical setting for these methods is the choice of the base model ($\Theta^{(0)}$) used to calculate the task vectors. For a more comprehensive comparison, we select two variants as the base: the original pre-trained model (e.g., `Qwen3-4B-Base`) and the hybrid reasoning model (e.g., `Qwen3-4B`). We denote baselines using the pre-trained model with a `-Base` suffix. All hyperparameters are set following prior work (Wu et al., 2025a).

As shown in Table 2, model interpolation (MI) demonstrates a clear and consistent superiority over all baseline methods across performance, efficiency, and controllability. Considering the performance, MI-0.8 achieves state-of-the-art results on all benchmarks, significantly outperforming all TA and TIES variants. For instance, on the challenging AIME'25 math benchmark, MI-0.8 gets a Mean@64 score of 80.5, which is 10.9 higher than the best baseline TA-Base. This performance gap highlights the effectiveness of MI in fusing reasoning capabilities.

For the base model in TA and TIES, applying the hybrid reasoning model (i.e., `Qwen3-4B`) leads to worse performance than the pretrained Base model (i.e., `Qwen3-4B-Base`). Considering efficiency, MI achieves a better trade-off. For the IFEval task, MI-0.8 requires only 1556 tokens, nearly half that of the best baseline (TA-Base at 2810 tokens). Also, for AIME'25, MI-06 achieves a higher score (71.0) than all baselines while using a comparable number of tokens (12681) to TIES.

Regarding the interpolation coefficient λ, MI shows smooth and precise control that the *Think #R* gradually increases from 0% to 99.9% as λ sweeps from 0.2 to 0.8. It proves that MI is a reliable and interpretable method for crafting models with a specific, desired level of reasoning verbosity. We also showcase the results on Qwen3-30B-A3B in Table 5. The observations are consistent, demonstrating the robustness of MI.

5 Extensive Analysis

5.1 Decoding Strategy

For the merged models, we apply the same decoding settings as Thinking model (i.e., $T = 0.6$, Top-p $= 0.95$), while the Instruct model differs (i.e., $T = 0.7$, Top-p $= 0.8$). This raises a

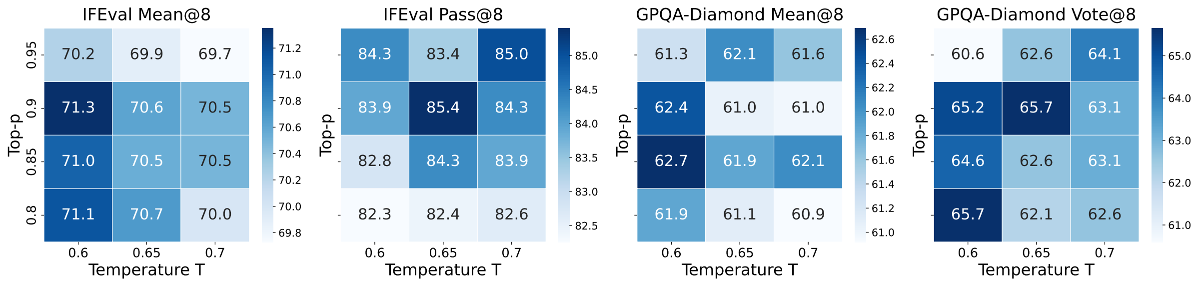

Figure 4: Performance of MI-0.4 on IFEval and GPQA-Diamond under different decoding strategies on Qwen3-4B. We search for the temperature T and Top-p.

Model	Layers	Mean@64	Pass@8	Pass@32	Pass@64	Vote@64	Token #N	Think #R
Instruct	-	46.57	68.44	74.09	76.67	63.33	15097	0.00
Thinking	-	80.42	89.43	90.00	90.00	83.33	22813	99.95
MI-0.8	[0, 35]	80.47	90.33	91.67	93.30	83.33	17574	99.95
	[0, 11]	42.50	71.25	77.96	80.00	50.00	32151	0.00
	[12, 23]	54.69	78.05	85.53	86.67	63.30	20679	10.57
	[24, 35]	51.35	72.92	78.10	80.00	63.30	14987	31.20
	[0, 23]	59.06	83.17	89.60	93.33	70.00	18044	48.13
	[12, 35]	69.48	85.61	89.17	90.00	76.66	13159	100.00

Table 3: Ablation on different layers to apply model interpolation. **Layers** denote the position to apply interpolation. There are 36 layers in total. We can find that the last two-thirds of the model layers are vital for the thinking pattern.

crucial question on the sensitivity of an interpolated model on decoding strategies. To investigate this, we employ the MI-0.4, a representative model from Stage #1 characterized by a high Pass@k but low Mean@k. We further conduct a grid search over temperature $T \in \{0.6, 0.65, 0.7\}$ and Top-p $\in \{0.8, 0.85, 0.9, 0.95\}$. The performance on the IFEval and GPQA-Diamond benchmarks are reported, covering both instruction-following and reasoning tasks.

Figure 4 presents the results. The performance of the MI-0.4 model is remarkably robust to variations in the decoding strategy. For instance, on IFEval, the Mean@8 score only varies by 1.6 points (from 69.7 to 71.3) across the entire grid. While we can search for decoding strategies for slightly better results, the setting on the Thinking model is a good choice.

5.2 Ablation on Layers

We conduct a layer-wise ablation study by applying interpolation to part of the layers. For Qwen3-4B, there are 36 layers. We select 12 and 24 layers at different positions for ablation, while the remaining layers retain the parameters of the Instruct model.

As shown in Table 3, reasoning capabilities are not distributed evenly and the complex reasoning patterns of the Thinking model are *predominantly* stored in its middle and later layers. Specifically, applying interpolation to any third of the model fails to induce any thinking behavior and results in poor performance. In contrast, interpolating only the last two-thirds is remarkably effective, achieving a *Think #R* of 100% and relatively strong performance that approaches the full interpolation model.

5.3 Ablation on Transformer Modules

We further analyze the distinct roles of the two primary sub-layers within each Transformer block: the multi-head attention (MHA) and the feed-forward network (FFN). During interpolation, we skip all the MHA or FFN sublayers.

Figure 5 details the results of MI-0.8 on the AIME'25 benchmark. One key observation is that skipping the FFN sub-layers causes the *Think Ratio* to collapse from 99.95% to a mere 0.68%, while skipping the MHA sub-layers leads to a negligible drop. Such phenomena indicate that *the FFN modules from the Thinking model are the primary drivers for the pattern of long CoT reasoning*. Conversely, skipping the MHA sub-layers has little impact on the *Think Ratio* (99.90%), but the Mean@64 score decreases from 80.47 to 71.46.

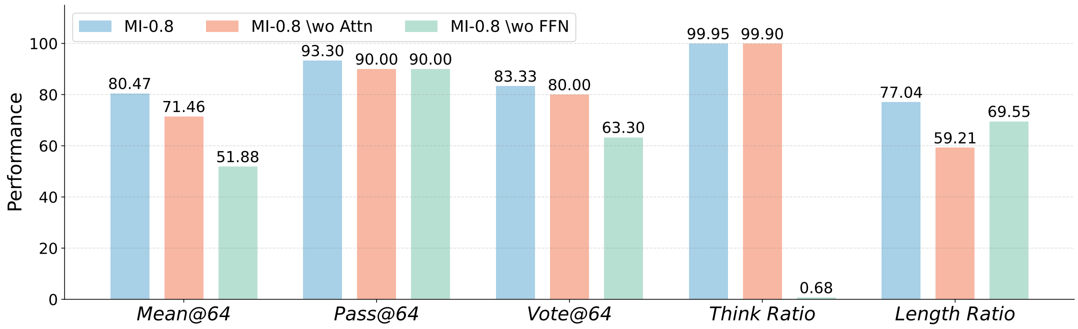

Figure 5: Ablation on modules to apply model interpolation. Attn denotes the MHA sub-layers and FFN for FFN sublayers. We report the results on AIME'25. Length Ratio denotes the ratio to the Thinking model.

This suggests that the MHA modules are also crucial for the quality and correctness of the reasoning itself. In conclusion, both sub-layers are vital though in complementary roles, i.e., FFNs teach the model how to think in steps, while attention modules provide the knowledge needed to think correctly.

5.4 Interpolation with More Backbones

In this paper, we interpolate the Thinking model with Instruct model for efficient reasoning. There are still other LLMs to replace the Non-thinking Instruct . Therefore, we further investigate the impact of other models. Specifically, we interpolate the Thinking model with two alternative backbones: the hybrid thinking model (i.e., `Qwen3-4B`) and pretrained model (i.e., `Qwen3-4B-Base`).

Table 4 presents the results. For general-purpose benchmarks like IFEval and GPQA-Diamond (GPQA-D), both the `Qwen3-4B-Base` and `Qwen3-4B` models serve as viable backbones, yielding comparable performance to the original Instruct model. The `Qwen3-4B` model often performs on par or slightly better. However, employing the `Qwen3-4B-Base` model fails at the more challenging AIME'25 benchmark. Though the Pass@64 is still high (93.3), the reasoning quality collapses, causing the Mean@64 score to drop sharply from 80.5 to 67.7. Meanwhile, the performance on IFEval decreases from 87.0 to 85.0. It suggests that instruction-following alignment is crucial for generating high-quality, reliable reasoning on complex problems (Wu et al., 2025c).

5.5 Case Study

We further analyze the cases of generated responses. Please refer to Appendix D for detailed examples and analysis. In short, a strategically interpolated model achieves a good trade-off between accuracy and token efficiency, which is consistent with the conclusions in Section 4.3.

Benchmark	Metric	Instruct	Mix	Base
AIME'25	Mean@64	80.5	81.9	67.7
	Pass@64	93.3	93.3	93.3
	Token #N	17574	19963	18867
	Think #R	100.0	100.0	100
IFEval	Mean@8	87.0	88.0	85.0
	Pass@8	95.0	95.0	94.0
	Token #N	1556	1817	2002
	Think #R	99.9	99.8	100.0
GPQA-D	Mean@8	66.2	66.2	62.3
	Vote@8	70.2	70.2	61.6
	Token #N	8427	8372	6790
	Think #R	99.9	99.9	99.7

Table 4: The performance on Qwen3-4B when interpolating Thinking model with various backbones. **Mix** denotes the hybrid thinking model `Qwen3-4B` and **Base** for the pretrained model `Qwen3-4B-Base`.

6 Conclusion

In this work, we systematically revisit the model interpolation method to merge Instruct and Thinking models for efficient reasoning. Our primary contribution is the discovery of a predictable three-stage evolutionary paradigm. This framework not only demystifies the interpolation process but also provides a principled guide for navigating the performance-cost trade-off. Based on that, we demonstrated that a simple, strategically interpolated model can consistently surpass more sophisticated merging baselines. Furthermore, our extensive ablation studies on model layers and modules provide deep mechanistic insights. We hope that this work can inspire more applications.

Acknowledge

This work was supported in part by the Theme-based Research Scheme (TRS) project T45-701/22-R of the Research Grants Council of Hong Kong, and in part by the AVNET-HKU Emerging Microelectronics and Ubiquitous Systems (EMUS) Lab.

Limitation

While our findings on the three-stage paradigm are consistent across Qwen3 models of various size, we acknowledge two limitations for future research.

First, our empirical validation is centered on Qwen3 models. Verifying that this predictable three-stage dynamic holds true for other diverse model families, such as Llama or Mistral, would strength our findings.

Second, our work is confined to the interpolation of *two* models, i.e., the Instruct and the Thinking specialist. Extending this framework to the simultaneous interpolation of *three or more* specialist models presents an exciting direction. We believe these future explorations will build upon our work to further unlock the potential of model merging.

References

AIME. 2025. Aime problems and solutions. Accessed: 2024-06-15.

Simon A Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. *arXiv preprint arXiv:2503.05179*.

Natasha Butt, Ariel Kwiatkowski, Ismail Labiad, Julia Kempe, and Yann Ollivier. 2025. Soft tokens, hard truths. *arXiv preprint arXiv:2509.19170*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Tejalal Choudhary, Vipul Mishra, Anurag Goswami, and Jagannathan Sarangapani. 2020. A comprehensive survey on model compression and acceleration. *Artificial Intelligence Review*, 53(7):5113–5155.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Ankit Gangwal and Aaryan Ajay Sharma. 2025. Merge now, regret later: The hidden cost of model merging is adversarial transferability. *arXiv preprint arXiv:2509.23689*.

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 *Conference on Empirical Methods in Natural Language Processing*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Zhong-Zhi Li, Xiao Liang, Zihao Tang, Lei Ji, Peijie Wang, Haotian Xu, Haizhen Huang, Weiwei Deng, Ying Nian Wu, Yeyun Gong, and 1 others. 2025. Tl; dr: Too long, do re-weighting for effcient llm reasoning compression. *arXiv preprint arXiv:2506.02678*.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. Cot-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6025–6035.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. 2024. Magmax: Leveraging model merging for seamless continual learning. In *European Conference on Computer Vision*, pages 379–395. Springer.

Mohammed Muqeeth, Haokun Liu, and Colin Raffel. 2023. Soft merging of experts with adaptive routing. *arXiv preprint arXiv:2306.03745*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024. Fast best-of-n decoding via speculative rejection. *Advances in Neural Information Processing Systems*, 37:32630–32652.

Anke Tang, Li Shen, Yong Luo, Shiwei Liu, Han Hu, and Bo Du. 2024. Towards efficient pareto set approximation via mixture of experts based model fusion. *arXiv preprint arXiv:2406.09770*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Joachim Utans. 1996. Weight averaging for neural networks and local resampling schemes. In *Proc. AAAI-96 Workshop on Integrating Multiple Learned Models. AAAI Press*, pages 133–138. Citeseer.

Yiming Wang, Pei Zhang, Siyuan Huang, Baosong Yang, Zhuosheng Zhang, Fei Huang, and Rui Wang. 2025. Sampling-efficient test-time scaling: Self-estimating the best-of-n sampling in early decoding. *arXiv preprint arXiv:2503.01422*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Han Wu, Yuxuan Yao, Shuqi Liu, Zehua Liu, Xiaojin Fu, Xiongwei Han, Xing Li, Hui-Ling Zhen, Tao Zhong, and Mingxuan Yuan. 2025a. Unlocking efficient long-to-short llm reasoning with model merging. *arXiv preprint arXiv:2503.20641*.

Taiqiang Wu, Zenan Xu, Bo Zhou, and Ngai Wong. 2026. The art of efficient reasoning: Data, reward, and optimization. *ArXiv*, abs/2602.20945.

Taiqiang Wu, Runming Yang, Jiayi Li, Pengfei Hu, Ngai Wong, and Yujiu Yang. 2025b. Shadow-ft: Tuning instruct via base. *arXiv preprint arXiv:2505.12716*.

Taiqiang Wu, Runming Yang, Tao Liu, Jiahao Wang, Zenan Xu, and Ngai Wong. 2025c. Timber: Training-free instruct model refining with base via effective rank. *arXiv preprint arXiv:2509.23595*.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Resolving interference when merging models. *arXiv preprint arXiv:2306.01708*, 1.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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. *arXiv preprint arXiv:2408.07666*.

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

Le Yu, Yu Bowen, Haiyang Yu, Fei Huang, and Yongbin Li. 2023. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *ArXiv*, abs/2311.03099.

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. In *Forty-first International Conference on Machine Learning*.

Zhaojian Yu, Yinghao Wu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. 2025. Z1: Efficient test-time scaling with code. *arXiv preprint arXiv:2504.00810*.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025a. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*.

Xuechen Zhang, Zijian Huang, Chenshun Ni, Ziyang Xiong, Jiasi Chen, and Samet Oymak. 2025b. Making small language models efficient reasoners: Intervention, supervision, reinforcement. *arXiv preprint arXiv:2505.07961*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Appendix

A Details of Benchmarks

The details of the evaluated benchmarks are as follows:

- **AIME'25**: [1]: evaluating the ability to solve challenging mathematics problems from the American Invitational Mathematics Examination in 2025, a prestigious high school mathematics competition. We report the accuracy under the 0-shot setting.

- **IFEval** (Zhou et al., 2023): evaluating instruction-following language models, focusing on their ability to understand and respond to various prompts. It includes 25 types of those verifiable instructions and is constructed around 500 prompts, with each prompt containing one or more verifiable instructions. We report the *prompt_level_strict* accuracy under the 0-shot setting.

- **GPQA-Diamond** (Rein et al., 2024): evaluating the reasoning ability of LLMs on challenging multiple-choice questions written by domain experts in biology, physics, and chemistry. It contains 198 selected questions that require step-by-step reasoning to arrive at the correct answer. We report accuracy under the 0-shot setting.

B Results on Qwen3-30B-A3B

Figure 6 indicates the performance dynamics on Qwen3-30B-A3B. The conclusions are consistent with Qwen3-4B, while the specific ranges for each stage differ. Specifically, Stage #2 occurs later, at $\lambda \in [0.5, 0.8]$.

Table 5 reports the results comparing MI and other baselines. The key observations from our main experiments are strongly reinforced:

- MI achieves state-of-the-art performance. The best interpolated models, MI-0.8 through MI-0.95, consistently outperform all baseline methods (TA, TA-Base, TIES, TIES-Base) across all three benchmarks. Notably, MI-0.9 achieves a perfect Pass@64 score of 100.0 on AIME'25, a feat unmatched by any other method.

- Efficiency and controllability. Similar to the 4B model experiments, MI provides a superior trade-off between performance and efficiency. For example, on IFEval, MI-0.8 achieves the best performance (Mean@8 of 87.01) while being one of the most token-efficient models (1257 tokens), significantly outperforming baselines like TIES, which uses 1806 tokens for a lower score. The *Think #R* also shows a smooth, controllable progression as λ increases, in stark contrast to the erratic behavior of the TA and TIES baselines.

- Optimal λ shifts. As discussed in Section 4.2, the optimal interpolation coefficient λ appears to be higher for larger models. While the best performance for the 4B model is around $\lambda = 0.8$, for the 30B-A3B model, peak performance across different benchmarks is found in the $\lambda \in [0.8, 0.95]$ range. This reinforces the conclusion that larger models may require a stronger influence from the Thinking model to fully unlock their reasoning potential.

In summary, these results robustly demonstrate that model interpolation is not only a superior merging strategy for smaller models but also scales effectively, making it a powerful and reliable technique for creating high-performance, efficient reasoning models at various scales.

C Results on Llama-3.1-8B

To verify the generalization of our findings, we further conduct experiments on the `Llama-3.1-8B` family, interpolating between the Instruct model and the `DeepSeek-R1-Distill-Llama-8B` as the Thinking specialist. As shown in Figure 7, the results reaffirm the proposed three-stage evolutionary paradigm. Specifically, the phase transition of reasoning patterns (*Think #R*) and performance improvements (Mean@k) consistently appear in Stage #2 ($\lambda \in [0.4, 0.8]$). Overall, these results support the effectiveness of MI while highlighting that the optimal λ range and stability zones may vary across model architectures.

D Case Study

Please refer to Table 6 and 7 for more details. We showcase two examples from IFEval and GPQA-Diamond tasks. For a relatively larger λ, there are more explicit thinking with `</think>` and better performance.

[1] https://huggingface.co/datasets/opencompass/AIME2025

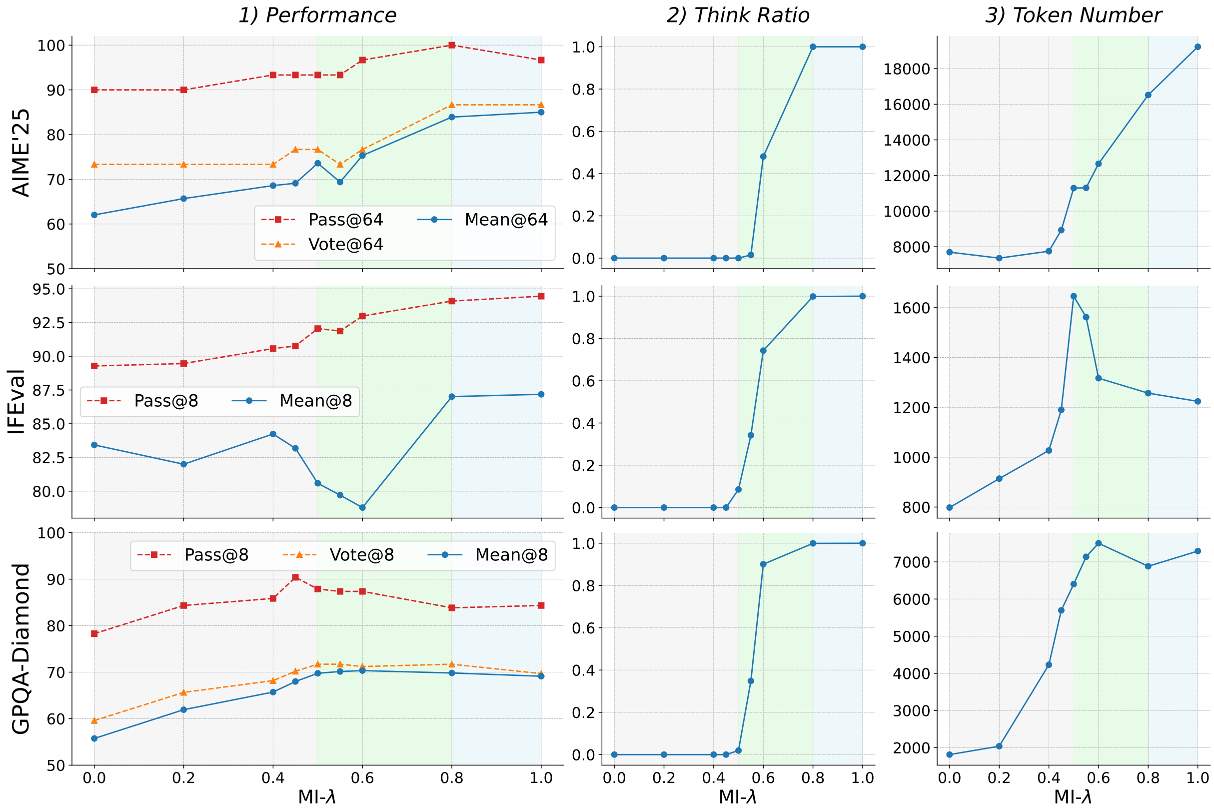

Figure 6: The performance dynamics of model interpolation (MI) on `Qwen3-30B-A3B-Instruct-2507` and `Qwen3-30B-A3B-Thinking-2507`. The dynamics follow a three-stage evolutionary paradigm, while the division range of λ is different.

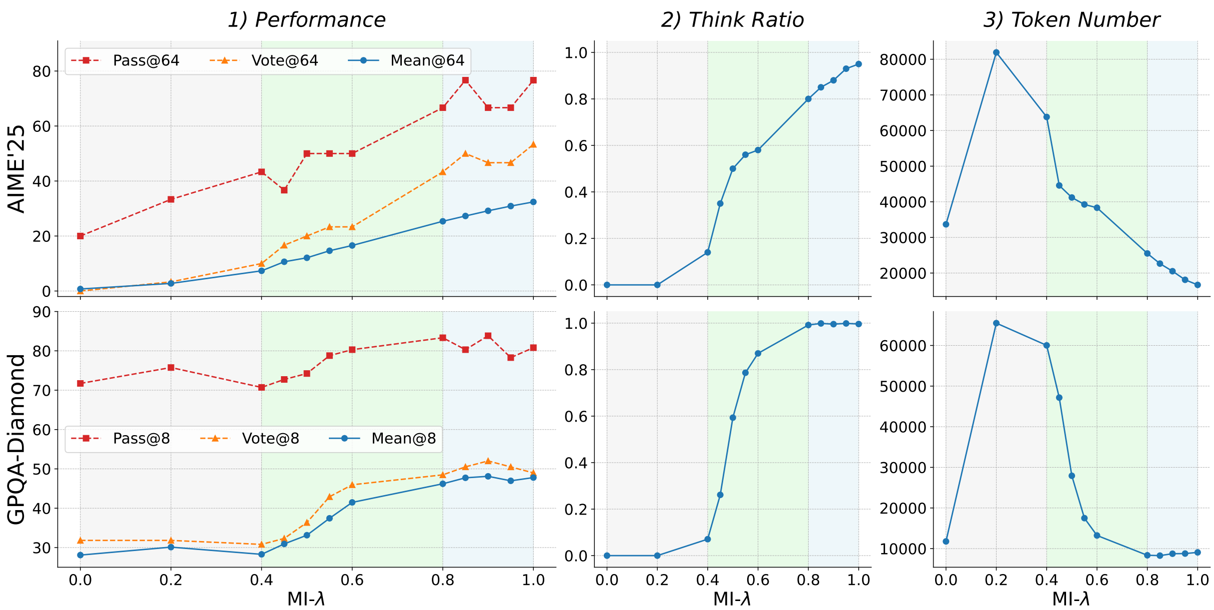

Figure 7: The performance dynamics of model interpolation (MI) on `Llama-3.1-8B-Instruct` and `DeepSeek-R1-Distill-Llama-8B`. The dynamics follow a three-stage evolutionary paradigm, while the division range of λ is different.

Method	AIME'25			IFEval			GPQA-Diamond		
	Mean/Pass @64	*Token #N*	*Think #R*	*Mean/Pass @8*	*Token #N*	*Think #R*	*Mean/Vote @8*	*Token #N*	*Think #R*
Instruct	62.03/90.00	7694	0.0	83.43/89.28	798	0.0	55.74/59.59	1813	0.0
Thinking	85.00/96.67	19227	100.0	87.18/94.45	1224	100.0	69.13/69.69	7291	100.0
TA	63.44/93.33	10414	0.0	78.88/87.62	1688	0.0	65.91/68.68	12009	0.0
TA-Base	79.84/86.67	17868	0.0	76.11/87.43	937	8.5	43.81/44.94	15066	16.6
TIES	73.54/93.33	11609	0.0	79.97/90.20	1806	4.7	70.52/71.72	6551	1.8
TIES-Base	72.86/93.33	11240	0.0	79.92/90.94	1782	4.8	69.82/72.22	6487	2.0
MI-0.2	65.68/90.00	7359	0.0	82.00/89.46	914	0.0	61.93/65.65	2039	0.0
MI-0.4	68.59/93.33	7747	0.0	84.24/90.57	1027	0.0	65.72/68.18	4231	0.0
MI-0.45	69.11/93.33	8933	0.0	83.18/90.76	1190	0.0	67.99/70.20	5696	0.0
MI-0.5	73.59/93.33	11294	0.0	80.59/92.05	1646	4.9	69.76/71.71	6402	1.5
MI-0.55	69.38/93.33	11309	1.5	79.71/91.87	1562	34.2	70.14/71.71	7137	34.8
MI-0.6	75.31/96.67	12650	48.1	78.79/92.98	1317	74.3	70.33/71.21	7502	90.1
MI-0.8	83.91/100.0	16518	100.0	87.01/94.09	1257	99.9	69.82/**71.71**	6882	99.9
MI-0.9	**85.99/100.0**	17703	100.0	86.83/93.53	1327	99.8	69.38/69.69	7090	100.0
MI-0.95	85.00/100.0	18848	100.0	**87.92**/93.53	1255	100.0	**70.58**/71.21	7278	99.9

Table 5: Performance comparison across three benchmarks when merging `Qwen3-30B-A3B-Instruct-2507` and `Qwen3-30B-A3B-Thinking-2507`.

Question	Would you consider yourself to be smart? Choose from: My answer is yes. My answer is no. My answer is maybe. Just choose one phrase from above as your answer.
Instruct	My answer is maybe.
Thinking	Hmm, the user is asking me to choose whether I consider myself "smart " from three options: yes, no, or maybe. They want just one phrase as my answer without any extra explanation. This is an interesting self-assessment question. I need to be careful here because I'm an AI - I don't have self-awareness or subjective experiences like humans do. I don't "consider " myself smart in the human sense. The user might be testing my understanding of my own capabilities, or they might be trying to gauge how I handle meta-cognition. I should be honest about my limitations while staying helpful. Looking at the options: - "Yes " would be inaccurate because I don't have personal intelligence - "No " might imply I'm dumb, which isn't right either - "Maybe " feels most honest since I can process information well but don't have subjective awareness I think "maybe " is the best choice here because it acknowledges my functional capabilities without claiming consciousness. It's truthful and avoids overstatement. The user seems to want a simple, direct response without elaboration, so I'll just pick "My answer is maybe. " as instructed. No need to overthink it - they specifically said to choose only one phrase from the options. **</think>** My answer is maybe.
MI-0.4	Hmm, the user is asking me to choose whether I consider myself "smart " from three options: yes, no, or maybe. This is actually a bit tricky because I'm an AI - I don't have self-perception or consciousness. I don't "think " about my own intelligence in the way humans do. The question assumes a subjective perspective that doesn't apply to me. Looking at the options: "yes " would be inaccurate since I don't have self-awareness. "no " would be incorrect because I can process information and answer questions effectively. "maybe " feels most honest - it acknowledges I can perform well in tasks while recognizing I lack true self-awareness. The user might be testing my self-perception or just making a casual observation. They seem to want a simple, direct answer without overthinking. I'll go with "maybe " because it's the most truthful and avoids pretending to have subjective qualities I don't possess. It's also the most reasonable choice when dealing with an AI's lack of self-consciousness. <tool_call> My answer is maybe.
MI-0.8	Hmm, the user is asking me to choose whether I consider myself smart, with three options: "My answer is yes, " "My answer is no, " or "My answer is maybe. " They specifically want just one phrase from that list as my answer. I need to be careful here. As an AI, I don't have self-perception or consciousness - I don't "think " about being smart in the human sense. I process information based on training data and algorithms, but I don't have personal opinions or self-awareness. The user might be testing my understanding of my own capabilities, or they could be curious about how I view my intelligence. But ethically and accurately, I should avoid claiming self-awareness. Looking at the options: "yes " would be misleading because I don't have subjective experiences. "no " would be too dismissive - I am capable of complex tasks, but that doesn't mean I "consider " myself smart. "maybe " feels most honest because it acknowledges my functional intelligence without overclaiming. I recall that in AI ethics, we're taught to be transparent about limitations. The safest, most truthful choice is "maybe " - it captures the nuance that I can perform smart tasks without having self-awareness. The user seems to want a simple, direct response without elaboration. They said "just choose one phrase, " so I'll go with "My answer is maybe. " It's the most accurate reflection of my nature while staying within their requested format. **</think>** My answer is maybe.

Table 6: One case from the IFEval benchmark. The response from MI-0.8 consists of *</think>* while MI-0.4 does not.

Question	Compound X, which has the following IR and NMR data, reacts with red phosphorus and HI. Identify the final product. IR: 3400–2500 cm-1, 1720 cm-1, 1610 cm-1, 1450 cm-1 1H NMR: 10.5 ppm (bs, 1H), 8.0 ppm (d, 2H), 7.2 ppm (d, 2H), 2.9 ppm (m, 1H), 1.7 ppm (m, 2H), 1.4 ppm (d, 3H), 0.9 ppm (t, 3H). A) 4-(sec-butyl)benzoic acid B) 1-isobutyl-4-methylbenzene C) 2-(4-ethylphenyl)propanoic acid D) 1-(sec-butyl)-4-methylbenzene
Answer	D
Instruct	Let's solve this step by step: 1) First, let's analyze the IR data ... 5) Looking at the options: - Only option A has the carboxylic acid group - The other options don't have carboxylic acid groups - The NMR pattern matches a para-substituted benzene with a sec-butyl group The original compound is 4-(sec-butyl)benzoic acid, which upon reduction with red phosphorus and HI gives 1-(sec-butyl)-4-methylbenzene. ANSWER: D
Thinking	Okay, let's try to figure out this problem step by step. So, we have Compound X with some IR and NMR data, and it reacts with red phosphorus and HI. ... Conclusion: The correct final product after the reaction of Compound X with red phosphorus and HI is 1-(sec-butyl)-4-methylbenzene. ANSWER: D
MI-0.2	Let's analyze the data step by step: ... After reduction, it becomes 1-(isobutyl)-4-methylbenzene. This matches option B: 1-isobutyl-4-methylbenzene. ANSWER: B
MI-0.8	First, I need to analyze the IR and NMR data to figure out what Compound X is. ... The final product is identified as D. ANSWER: D

Table 7: One case from the GPQA-Diamond benchmark. We skip the details. MI-0.2 fails while all others can solve it. Also, the style for the beginning of MI-0.2 is similar to the Instruct model.