System 1&2 Synergy via Dynamic Model Interpolation

Chenxu Yang[1,2], **Qingyi Si**[3], **Chong Tian**[4], **Xiyu Liu**[1,2], **Dingyu Yao**[1,2], **Chuanyu Qin**[1,2], **Zheng Lin**[1,2], **Weiping Wang**[1], **Jiaqi Wang**[3]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]JD.COM
[4]MBZUAI
{yangchenxu,linzheng}@iie.ac.cn

Abstract

Training a unified language model that adapts between intuitive System 1 and deliberative System 2 remains challenging due to interference between their cognitive modes. Recent studies have thus pursued making System 2 models more efficient. However, these approaches focused on output control, limiting what models produce. We argue that this paradigm is misaligned: output length is merely a symptom of the model's cognitive configuration, not the root cause. In this work, we shift the focus to capability control, which modulates *how models think* rather than *what they produce*. To realize this, we leverage existing Instruct and Thinking checkpoints through dynamic parameter interpolation, without additional training. Our pilot study establishes that linear interpolation yields a convex, monotonic Pareto frontier, underpinned by representation continuity and structural connectivity. Building on this, we propose **DAMI** (**D**ynA**m**ic **M**odel **I**nterpolation), a framework that estimates a query-specific Reasoning Intensity $\lambda(q)$ to configure cognitive depth. For training-based estimation, we develop a preference learning method encoding accuracy and efficiency criteria. For zero-shot deployment, we introduce a confidence-based method leveraging inter-model cognitive discrepancy. Experiments on five mathematical reasoning benchmarks demonstrate that DAMI achieves higher accuracy than the Thinking model while remaining efficient, effectively combining the efficiency of System 1 with the reasoning depth of System 2.

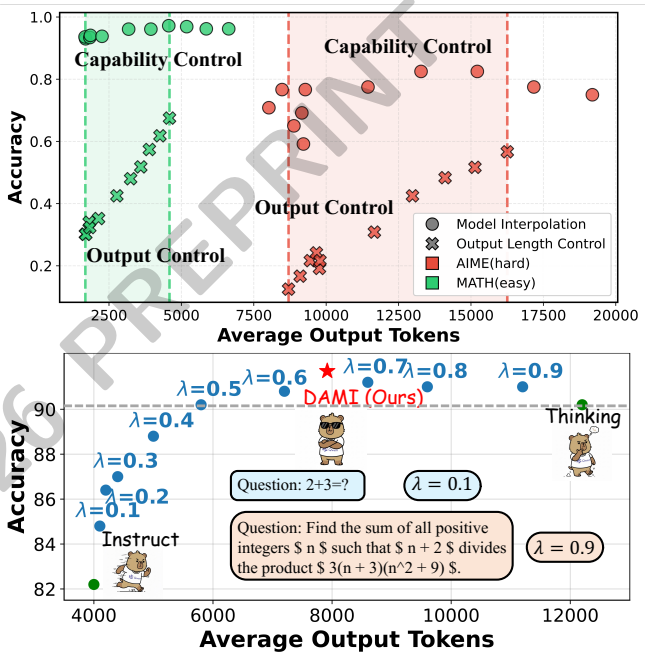

Figure 1: Capability control outperforms output control across different efficiency levels on both tasks. Linear interpolation between Instruct and Thinking yields a monotonic Pareto frontier.

1 Introduction

The cognitive evolution of Large Language Models (LLMs) has established a functional dichotomy: lightweight "System 1" models excel at rapid, intuitive responses, while "System 2" models such as OpenAI's o1 [OpenAI, 2024] and DeepSeek-R1 [DeepSeek-AI *et al.*, 2025] achieve superior reasoning through extended Chain-of-Thought (CoT) [Li *et al.*, 2025]. Ideally, a unified model would seamlessly adapt between these modes, responding swiftly to simple queries while engaging deep deliberation for complex problems [Team, 2025]. However, training such an adaptive model remains an open challenge due to the zero-sum nature of training data [Shukor *et al.*, 2025]. The verbose, self-reflective style of System 2 reasoning conflicts with the concise responses of System 1, causing mutual interference during joint optimization. Consequently, state-of-the-art LLMs typically release separate Thinking and Instruct checkpoints rather than a single unified model [Bai *et al.*, 2025].

Given this difficulty, the community has pursued a more tractable alternative: making System 2 models more efficient. This pursuit has spawned various approaches, from training-free methods like token budgeting [Han *et al.*, 2025], early exit [Yang *et al.*, 2025c], and CoT compression [Xia *et al.*, 2025], to training-based methods employing reinforcement learning (RL) with length penalties [Arora and Zanette, 2025; Su and Cardie, 2025] or trajectory pruning [Dai *et al.*, 2025;

Hou *et al.*, 2025]. Despite their diversity, these approaches share a common paradigm: **output control**, which limits *what* models produce. We argue that this framing is misaligned: output length is merely a symptom of the model's cognitive configuration, not the root cause of inefficiency. As illustrated in Figure 1, constraining generation with token budgets risks truncating the reasoning process, resulting in incomplete chains of thought that severely degrade performance on challenging tasks.

In this work, we shift the focus from output control to **capability control**, which modulates *how models think* rather than *what they produce*. Our key observation is that by interpolating parameters between existing System 1 and System 2 checkpoints, we can smoothly adjust the model's cognitive intensity without additional training. As shown in Figure 1 (upper), under comparable efficiency constraints, capability control consistently outperforms output control by producing naturally concise yet complete solutions.

A natural question arises: does parameter interpolation yield predictable behavior? Our pilot experiments in Section 3.1 provide affirmative evidence. We find that increasing the interpolation coefficient λ (where λ=0 corresponds to Instruct and λ=1 to Thinking) produces a smooth, monotonic Pareto frontier between accuracy and efficiency, as shown in Figure 1 (lower), with continuous transitions in representation space. This predictability is grounded in the structural similarity between models: cosine similarity exceeds 0.99 across all layers, indicating that both models satisfy Linear Mode Connectivity [Frankle *et al.*, 2020] and reside within the same optimization basin. These findings establish parameter interpolation as a principled mechanism for capability control.

Building on this foundation, we propose **DAMI** (**D**yn**A**mic **M**odel **I**nterpolation), a framework that dynamically estimates a query-specific Reasoning Intensity $\lambda(q) \in [0, 1]$ to configure the model's cognitive depth for each input. We explore two complementary methods tailored for different deployment scenarios: DAMI-Pref, a preference learning method that jointly encodes accuracy and efficiency criteria, is suited for accuracy-critical applications where in-domain training data is available; DAMI-Conf, a confidence-based method that leverages inter-model cognitive discrepancy, enables rapid deployment across diverse domains without requiring any training data. Experiments on five mathematical reasoning benchmarks show that DAMI improves accuracy by 1.6–3.4% while reducing token consumption by 29–40% compared to the Thinking model, outperforming static merging, early-exit, and routing baselines.

Our contributions are summarized as follows:

- We advocate a paradigm shift from output control to capability control for efficient reasoning, and identify dynamic model interpolation as a low-cost realization that leverages existing checkpoints, bypassing the difficulty of training System1&2 unified models
- We propose DAMI with two complementary $\lambda(q)$ estimation methods: preference-based learning for data-available scenarios and confidence-based inference for zero-shot deployment.
- Extensive experiments demonstrate that DAMI consistently establishes superior Pareto frontiers across diverse benchmarks and model families.

2 Related Work

2.1 Efficient Reasoning

The inherent trade-off between System 1's efficiency and System 2's accuracy has motivated a growing body of research on efficient reasoning, which can be categorized into single-model optimization and multi-model collaboration. Single-model approaches include early-exit methods that terminate generation upon sufficient confidence [Yang *et al.*, 2025c; Dai *et al.*, 2025], CoT compression that removes redundant reasoning steps [Xia *et al.*, 2025], adaptive reasoning that adjusts output length based on query complexity [Luo *et al.*, 2025; Shen *et al.*, 2025], and other trajectory-revised methods [Yang *et al.*, 2025b]. While adaptive methods share our motivation, they rely on RL with length penalties, incurring high training costs and potential performance degradation on difficult problems. Despite their diversity, these single-model approaches predominantly follow the output control paradigm, constraining *what* models produce. Multi-model approaches leverage complementary capabilities across models. Task decomposition methods delegate simple steps to lightweight models while reserving complex reasoning for stronger ones [Akhauri *et al.*, 2025; Fan *et al.*, 2025]. LLM Routing dynamically dispatches queries to suitable models from a pool [Ong *et al.*, 2025; Zhang *et al.*, 2025]. Model merging statically integrates parameters of different models to combine their strengths [Wu *et al.*, 2025a; Yao *et al.*, 2025]. **Our work bridges routing and merging by enabling continuous, query-adaptive interpolation** rather than discrete model selection or static merging.

2.2 Model Merging

Model merging integrates parameters from multiple specialized models into a unified model without requiring original training data. It has been applied to mitigate catastrophic forgetting in continual learning [Schumann *et al.*, 2023], integrate knowledge in multi-task learning [Yang *et al.*, 2024; Ilharco *et al.*, 2023], and aggregate updates in federated learning [Wang *et al.*, 2020]. Recent work has introduced merging into efficient reasoning, with leading LLM teams releasing merged models that balance reasoning depth and efficiency [Wu *et al.*, 2025b; Wang *et al.*, 2023; Yang *et al.*, 2025a]. However, existing approaches apply fixed merging coefficients across all queries. To our knowledge, we are the first to propose query-aware dynamic merging between System 1 and System 2 models, enabling fine-grained integration of their complementary strengths for adaptive control over reasoning depth.

3 Methodology

3.1 Why Interpolation Enable Capability Control

Before formulating our dynamic framework, we first validate that parameter interpolation can serve as a continuous and predictable control mechanism for reasoning. We conduct a

pilot study using static linear interpolation between the Instruct model $\Theta^{(\mathrm{I})}$ and the Thinking model $\Theta^{(\mathrm{T})}$ with a fixed coefficient $\lambda \in [0, 1]$ across the test set.

$$\Theta^{(\mathrm{M})} = \lambda\Theta^{(\mathrm{T})} + (1 - \lambda)\Theta^{(\mathrm{I})} \tag{1}$$

Performance Monotonicity. As shown in Figure 1, increasing λ from 0 to 0.9 yields a smooth improvement in accuracy alongside a predictable increase in token consumption. Crucially, this trajectory forms a convex Pareto frontier where intermediate interpolations can outperform both endpoints: achieving higher accuracy than the Instruct model while requiring fewer tokens than the Thinking model. This confirms that λ serves as a reliable proxy for reasoning depth, enabling fine-grained efficiency-accuracy trade-offs.

Representation Continuity. What underlies this behavioral smoothness? We analyze the hidden states of ⟨think⟩ token using Principal Component Analysis (PCA) across 50 samples on the MATH-500. As shown in Figure 2(a), representations of interpolated models form a continuous trajectory connecting the Instruct and Thinking endpoints. The first principal component strongly correlates with λ (r=0.974), indicating that interpolation induces a gradual cognitive transition rather than discrete jumps in the representation space.

Structural Connectivity. This representational continuity is grounded in the geometric properties of the parameter space. Figure 2(b) shows that the Instruct and Thinking models maintain high cosine similarity (>0.99) across all transformer layers, satisfying the Linear Mode Connectivity (LMC) property [Frankle *et al.*, 2020; Zhan *et al.*, 2025]. This indicates that both models reside within the same optimization basin, ensuring that interpolation traverses a valid path in the loss landscape. Notably, the L2 distance peaks in middle-to-deep layers, suggesting that System 2 capabilities are encoded through targeted modifications in these regions rather than global parameter shifts.

Key Insight. These findings establish a causal chain: structural alignment in parameter space → continuous transitions in representation space → monotonic Pareto frontier in performance. This connectivity transforms reasoning resource allocation into a well-posed prediction problem, where estimating the optimal $\lambda(q)$ directly determines model behavior. These findings jointly support interpolation validity, performance monotonicity, and representation continuity. Together, these properties ensure that model interpolation behaves predictably and monotonically, providing the theoretical foundation for fine-grained, query-level dynamic merging. This validates our core premise: λ can serve as a reliable control knob for reasoning intensity, enabling the adaptive framework we introduce next.

3.2 The Dynamic Interpolation Framework

While Section 3.1 validates that parameter interpolation provides a reliable control mechanism, a fixed λ is inherently suboptimal since different queries require different reasoning depths. DAMI addresses this by making λ query-dependent:

$$\Theta^{(\mathrm{M})}(q) = \lambda(q)\Theta^{(\mathrm{T})} + (1 - \lambda(q))\Theta^{(\mathrm{I})} \tag{2}$$

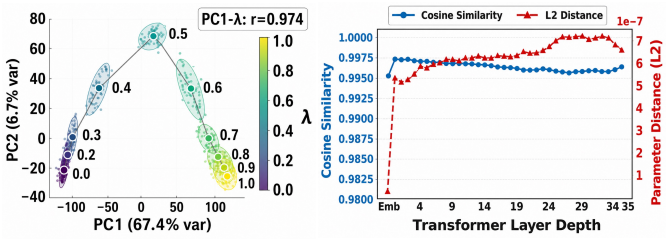

Figure 2: The monotonicity and continuity of reasoning intensity.

We term $\lambda(q) \in [0, 1]$ the **Reasoning Intensity**, which modulates the cognitive depth for each input. The key technical challenge is estimating the optimal $\lambda(q)$. We explore two complementary methods: **DAMI-Pref**, a preference learning approach that jointly encodes accuracy and efficiency criteria, suited for accuracy-critical applications where in-domain training data is available; and **DAMI-Conf**, a confidence-based approach that leverages inter-model cognitive discrepancy, enabling rapid deployment across diverse domains without requiring any training data.

3.3 Training-based Estimation of λ

Baseline: Regression-based Estimation

A straightforward solution is to train a lightweight router model R_ϕ to regress a pre-computed target $\lambda^*(q)$. We profile each query q across J discrete coefficients $\Lambda = \{l_1, \ldots, l_J\}$, measuring accuracy via N-sample evaluation:

$$\mathrm{Acc}^{l_j}(q) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}(\mathcal{M}_n^{l_j}(q), a) \tag{3}$$

where $\mathcal{M}_n^{l_j}(q)$ is the response from a model merged with coefficient l_j, and $\mathbb{I}(\cdot, \cdot)$ is a binary evaluation function. The target $\lambda^*(q)$ is defined as the minimum coefficient achieving the highest accuracy, and the router is trained with MSE loss:

$$\mathcal{L}(\phi) = \frac{1}{|\mathcal{D}_{\text{train}}|}\sum_{q \in \mathcal{D}_{\text{train}}} (R_\phi(q) - \lambda^*(q))^2 \tag{4}$$

However, this approach suffers from two weaknesses: (1) point-wise labels $\lambda^*(q)$ are inherently noisy due to sampling variance, and (2) regression requires large-scale data for robust generalization. These limitations motivate our preference-based method.

DAMI-Pref: Preference-based Estimation

Instead of regressing noisy labels, we recast λ estimation as a preference learning problem. A lightweight reward model R_ψ is trained to score the relative efficacy of different coefficients, offering three advantages: (1) pairwise comparisons are more robust to noise, (2) preference criteria can jointly encode accuracy and efficiency, and (3) the method requires substantially less training data.

Preference Data Construction. We construct policy performance tuples $P(q, l_j) = \langle l_j, \mathrm{Acc}(q, l_j), \mathrm{Cost}(q, l_j)\rangle$,

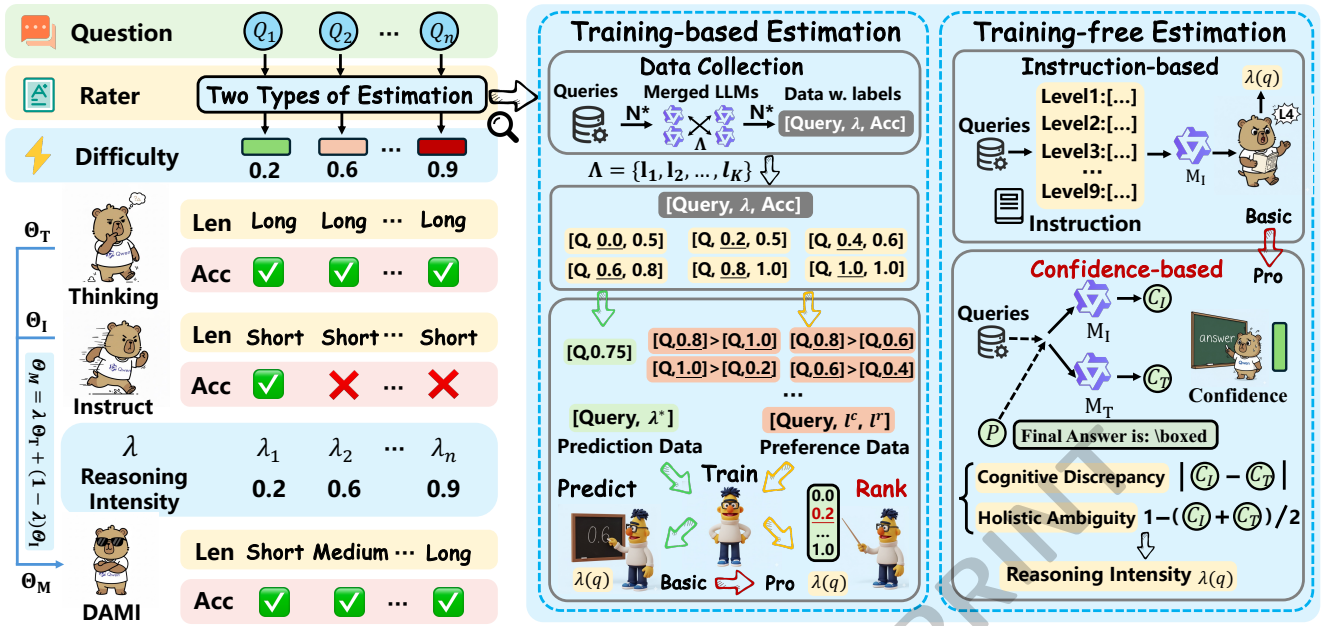

Figure 3: An overview of our DAMI method. DAMI estimates query difficulty through either preference learning (training-based) or confidence signals (training-free), then dynamically adjusts the Reasoning Intensity $\lambda(q)$ to configure cognitive depth accordingly.

where Cost is the average token count. Preference pairs are generated according to a hierarchical criterion:

$$P_i \succ P_j \iff \begin{cases} \text{Acc}_i > \text{Acc}_j + \delta_{\text{acc}} \\ |\text{Acc}_i - \text{Acc}_j| \le \delta_{\text{acc}} \wedge \text{Cost}_i < \text{Cost}_j \end{cases} \tag{5}$$

where δ_{acc} is a small tolerance threshold to account for sampling noise. This criterion prioritizes accuracy while using efficiency as a tiebreaker, directly reflecting the practical goal of achieving high performance at minimal cost. Pairs without clear preference are discarded to ensure signal quality. This process is applied to all pairs to construct the final dataset:

$$\mathcal{D}_{\text{pref}} = \{(q, l_c, l_r) \mid P(q, l_c) \succ P(q, l_r)\} \tag{6}$$

Optimization and Inference. The reward model is optimized via binary cross-entropy loss:

$$\mathcal{L}(\psi) = -\mathbb{E}_{(q,l_c,l_r)\sim\mathcal{D}_{\text{pref}}}[\log \sigma(R_\psi(q, l_c) - R_\psi(q, l_r))] \tag{7}$$

3.4 Training-free Estimation of λ

Baseline: Prompt-based Estimation

A straightforward approach is to leverage the model's in-context learning for difficulty assessment. We design a zero-shot prompt that instructs the Instruct model to rate query difficulty on a scale from 1 to 9, then linearly scale to obtain $\lambda(q)$. However, this method depends heavily on instruction-following capability and generalizes poorly across model families.

DAMI-Conf: Confidence-based Estimation

Considering the above drawbacks, we propose to derive $\lambda(q)$ from intrinsic confidence signals, offering two advantages: model-agnostic signals that generalize across architectures, and continuous-valued output for fine-grained control.

Confidence Extraction. Following [Yang *et al.*, 2025c], we define confidence as the geometric mean of maximum token probabilities over the answer sequence $\boldsymbol{a} = (a_1, \ldots, a_n)$:

$$\mathcal{C}(\mathcal{M}, q) = \left(\prod_{t=1}^{n} \max_{a_t \in \mathcal{V}} \text{softmax}(\mathcal{M}(q, i, \boldsymbol{a}_{<t}))_t \right)^{1/n} \tag{8}$$

where i is an answer-inducing prompt like "Final answer is: \boxed{". We compute $\mathcal{C}^I(q)$ and $\mathcal{C}^T(q)$ for the Instruct and Thinking models respectively.

Dual-Signal Fusion. We construct $\lambda(q)$ from two complementary signals that capture different aspects of query difficulty. *Holistic Ambiguity* measures overall system uncertainty. When neither model is confident, the query is likely challenging and requires deeper reasoning:

$$S_{\text{ambi}}(q) = 1 - \frac{\mathcal{C}^I(q) + \mathcal{C}^T(q)}{2} \tag{9}$$

Cognitive Discrepancy quantifies the disagreement between models by measuring the difference in their confidence. A high discrepancy suggests that the two models employ different strategies for addressing the query, indicating potential ambiguity that requires extensive reasoning to resolve:

$$S_{\text{dis}}(q) = |\mathcal{C}^I(q) - \mathcal{C}^T(q)| \tag{10}$$

The fused signal $S_{\text{final}} = S_{\text{ambi}} + S_{\text{dis}}$ is transformed via calibrated sigmoid to yield Reasoning Intensity:

$$\lambda(q) = \sigma\left(\frac{S_{\text{final}}(q) - \mu}{\tau}\right) \tag{11}$$

where μ and τ control the decision boundary and sensitivity.

Method	GSM8K				MATH-500				AMC				AIME24				AIME25				Overall		
	Acc↑ 2×	Tok↓ #N	CR↓ %	Think #R	Acc↑ 2×	Tok↓ #N	CR↓ %	Think #R	Acc↑ 8×	Tok↓ #N	CR↓ %	Think #R	Acc↑ 8×	Tok↓ #N	CR↓ %	Think #R	Acc↑ 8×	Tok↓ #N	CR↓ %	Think #R	Acc↑ -	CR↓ %	Think #R
Qwen3-4B-2507-Thinking/Qwen3-4B-2507-Instruct																							
Instruct	94.3	349	23.1	0	93.1	1741	26.1	0	93.8	3034	27.4	0	61.7	8938	46.5	0	49.2	8384	39.3	0	78.4	37.5	0
Thinking	**95.5**	1510	100	100	96.1	6668	100	100	99.4	11076	100	100	78.3	19241	100	100	71.7	21340	100	100	88.2	100	100
DEER	**95.5**	1059	70	100	95.4	6125	92	100	**100**	10084	91	100	83.3	18065	94	100	76.7	19287	90	100	90.2	87	100
TA	95	1225	81	100	**96.5**	4671	70	100	**100**	7760	70	100	78.3	13515	70	98	73.3	15546	73	98	88.6	73	99
TIES	94	363	24	0	94.8	1899	28	0	95.6	3012	27	0	68.3	8141	42	0	51.7	8419	39	0	80.9	32	0
MI-03	94	383	25	0	95.3	1933	29	0	96.2	3381	31	0	67.5	9708	50	0	51.7	9280	43	0	81	36	0
MI-05	95.1	1084	72	83	96.1	3213	48	72	96.2	5138	46	61	73.3	9321	48	50	65.8	10533	49	54	85.3	53	64
MI-07	**95.4**	1222	81	100	**96.4**	4617	69	100	**100**	7501	68	100	**82.5**	13993	73	100	75	16086	75	98	89.9	73	100
Routing	94.5	430	28	2.7	**96.4**	5435	82	53.8	**100**	10356	93	85	76.7	19861	103	100	73.3	21035	99	100	88.2	81	68.3
DAMI-Pred	94.4	365	24	0	95.6	1886	28	0	95	3477	31	0	73.3	7703	40	0	53.3	8776	41	0	83.3	33	0
DAMI-Prompt	**95.5**	1028	68	59	95	2832	42	27	95.6	4715	43	30	80	10648	55	76	75	13009	61	88	88.2	54	56
DAMI-Pref	94.5	843	56	25	**96.4**	4315	65	64	**98.8**	7872	71	86	**85**	15412	80	95	**83.3**	17488	82	92	**91.6**	71	72
DAMI-Conf	94.5	856	57	41	96.2	2954	44	34	**98.8**	5891	53	55	**85**	13485	70	79	**79.2**	16412	77	94	**90.7**	60	61
DeepSeek-R1-Distill-Qwen-7B/Qwen2.5-Math-7B																							
Instruct	55.3	1007	-	0	52	1268	-	0	42.5	1463	-	0	20	1470	-	0	3.3	1104	-	0	34.6	-	0
Thinking	89.6	1886	100	100	86	5881	100	100	**87.5**	8600	100	100	43.3	17887	100	100	**30**	21681	100	100	67.3	100	100
DEER	89.5	873	46	100	88.8	3140	53	100	82.5	7398	86	100	**45**	13945	78	100	**30**	15941	74	100	67.2	67	100
TA	90.7	769	41	99	88.8	2670	45	97	78.1	4319	50	96	39.2	9396	53	92	25.8	11189	52	83	64.5	48	93
TIES	86.5	346	18	0	80.2	827	14	0	58.1	1263	15	0	16.7	1739	10	0	10	1827	8	0	50.3	13	0
MI-03	85.5	379	20	0	76.8	1501	26	0	57.5	1817	21	0	23.3	3459	19	0	16.7	4485	21	0	52	21	0
MI-05	**91.9**	617	33	71	83.6	2338	40	13	72.5	4462	52	8	20	15330	86	0	20	10665	49	0	57.6	52	18
MI-07	91.2	902	48	99	88.2	3243	55	95	85	4867	57	92	36.7	18693	105	50	30	17169	79	60	66.2	69	79
Routing	64.5	1045	55	0	63.6	1303	22	4.4	50	1425	17	5	25	1875	10	20	6.7	2851	13	10	42	24	7.9
DAMI-Pred	83.5	410	22	0	79.6	1823	31	0	62.5	3103	36	20	26.7	13209	74	31	13.3	13813	64	26	53.1	45	15.4
DAMI-Prompt	82.1	431	23	0	76.2	1657	28	0	55	1697	20	0	20	3962	22	0	10	4962	23	0	48.7	23	0
DAMI-Pref	**92.4**	1121	59	48	**89.4**	3952	67	75	**88.8**	5819	68	88	**45**	14561	81	92	**32.5**	16837	78	94	**69.6**	71	79.4
DAMI-Conf	91.8	1050	56	55	**89**	3526	60	62	**87.5**	5299	62	78	**44.2**	13835	77	85	**30**	16174	75	90	**68.9**	66	74

Table 1: Experimental results on two model pairs (Qwen3-4B and Qwen2.5-7B) across five mathematical reasoning benchmarks. "Acc" denotes accuracy, "Tok" denotes token count, "CR" denotes compression rate, and "Think" denotes thinking ratio. ↑/↓ indicate that higher/lower values are better. The top-2 best results are highlighted in **bold**. The result is statistically significant with p-value < 0.05.

4 Experiments

4.1 Experimental Setup

Models and Datasets. We conduct experiments on the following two pairs of open-source models: (Qwen3-4B-2507-Thinking, Qwen3-4B-2507-Instruct) and (DeepSeek-R1-Distill-Qwen-7B, Qwen2.5-Math-7B). The evaluation benchmarks primarily consist of mathematical reasoning tasks, encompassing five specific datasets ranging from simple to difficult: GSM8K [Cobbe *et al.*, 2021], MATH-500 [Hendrycks *et al.*, 2021], AMC 2023, AIME 2024, and AIME 2025. For the training of the router model and reward model, we select training data from DeepMath [He *et al.*, 2025], which is out-of-distribution (OOD) relative to the evaluation sets. Since this dataset inherently contains difficulty labels, we first conduct an initial screening based on these labels, selecting an average of 1,000 samples across various difficulty levels for constructing the training data described in Section 3.2.

Baselines. We compare our methods (DAMI-Pref, DAMI-Conf) against three categories of approaches: (1) **output control methods**, including the early-exit approach DEER [Yang *et al.*, 2025c]; (2) **static capability control methods**, including Task Arithmetic (TA) [Ilharco *et al.*, 2023], TIES-Merging (TIES) [Yadav *et al.*, 2023], and Model Interpolation (MI) [Wu *et al.*, 2025b], which apply fixed merging coefficients across all queries; and (3) **dynamic capability control methods**, including prompt-based LLM Routing, as well as two strong dynamic merging baselines introduced in this work (DAMI-Pred, DAMI-Prompt). We also include the two original models (Instruct, Thinking) as reference endpoints. The hyperparameters for all baseline methods are set strictly according to the original papers.

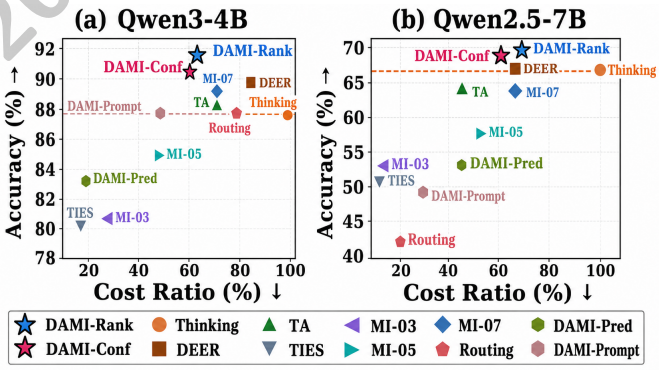

Figure 4: Accuracy-efficiency scatter plot aggregated over five benchmarks. DAMI methods (starred) achieve the best accuracy-efficiency trade-offs on both Qwen3-4B and Qwen2.5-7B.

Metrics and Implementations. We selected *Accuracy* (**Acc n×**), *Token Number* (**Tok #N**), and *Thinking Ratio* (**Think #R**) as our evaluation metrics. **Acc n×** denotes the final answer accuracy, and n × represents the number of sampling rounds conducted for evaluation. **Tok #N** indicates the average number of generated tokens per sample, serving as a measure of computational cost. **Think #R** is defined as the per-

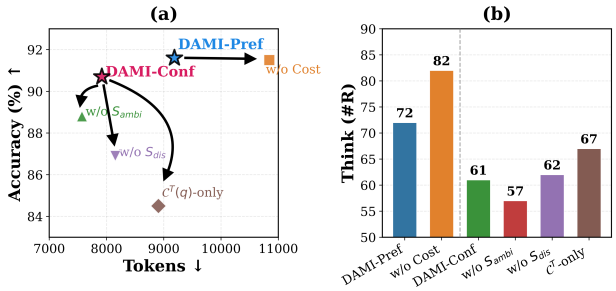

Figure 5: Ablation study results on the Qwen3-4B model pair.

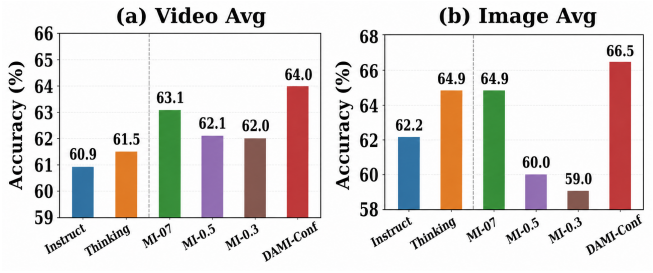

Figure 7: Generalization to multimodal tasks of DAMI.

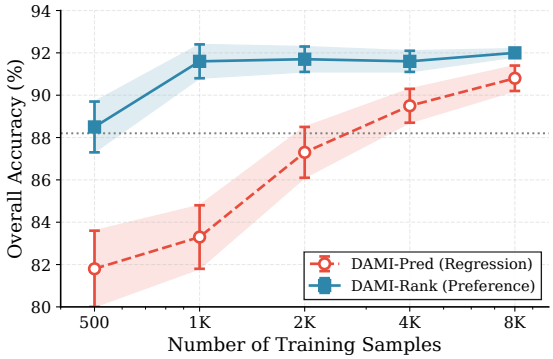

Figure 6: Effect of training data size on training-based λ estimation.

centage of responses containing the ⟨/think⟩ token, quantifying the prevalence of explicit CoT reasoning. Given the limited sample sizes in the AMC 2023, AIME 2024, and AIME 2025 datasets, we perform 8 sampling rounds per instance and average the results across all metrics to ensure statistical stability and reliability. Across all models, we consistently apply the hyperparameters $T = 0.6$ and Top-p$= 0.95$. For confidence-based estimation, we set $\mu = 0.3$ and $\tau = 0.3$.

4.2 Experimental Results

Overall Performance. As shown in Table 1, DAMI-Pref and DAMI-Conf consistently achieve the best performance across both model families. On Qwen3-4B pair, DAMI-Pref attains 91.6% accuracy (+3.4% absolute) while reducing token consumption by 29% compared to vanilla Thinking. DAMI-Conf further improves efficiency with 40% token reduction while maintaining 90.7% accuracy (+2.5% absolute). Figure 4 visualizes the overall accuracy-efficiency trade-off: DAMI-Pref and DAMI-Conf consistently occupy the Pareto-optimal region, dominating both output control methods and static capability control methods across both model pairs.

Comparison of Strong Dynamic Merging Baselines. The DAMI-Pred consistently underestimates required reasoning intensity, achieving only 83.3% accuracy versus 91.6% for DAMI-Pref on Qwen3-4B. The DAMI-Prompt performs reasonably on Qwen3-4B but degrades severely on 7B, as Qwen2.5-Math-7B lacks robust instruction-following for difficulty assessment. In contrast, DAMI-Pref benefits from pairwise comparisons that are robust to label noise, while DAMI-Conf leverages model-agnostic confidence signals that generalize across architectures.

Comparison with SoTA Methods. DEER achieves comparable accuracy to DAMI-Pref on Qwen3-4B but consumes more tokens and risks truncated reasoning. LLM Routing suffers from its discrete nature: on Qwen2.5-7B, it collapses to 42.0% accuracy due to the weak System 1 model, while our continuous interpolation maintains robust performance. Static merging methods (TA, TIES, MI-λ) cannot adapt to varying difficulty distributions. While MI-07 achieves strong results on Qwen3-4B, our dynamic approach consistently outperforms the best static configuration across benchmarks.

Analysis of Thinking Ratio. The Thinking Ratio reveals adaptive resource allocation. On simpler benchmarks like GSM8K, DAMI-Pref and DAMI-Conf activate deep thinking for only 25-41% of queries on Qwen3-4B. As difficulty increases toward AIME24/25, the ratio rises to 79-95%. This adaptive behavior distinguishes our approach from static methods, which either overthink simple queries (Thinking, TA) or underthink difficult ones (TIES, MI-03). The smooth variation demonstrates that DAMI successfully calibrates reasoning depth to query complexity.

4.3 Analysis

Ablation study. Figure 5 presents ablation results on the Qwen3-4B model pair, examining the contribution of key design components. For DAMI-Pref, removing the cost-based tiebreaker (-Cost) slightly improves accuracy but increases token consumption by 18.0%. This confirms that the hierarchical preference criterion effectively balances accuracy and efficiency. For DAMI-Conf, both signals contribute to the final performance. Removing Cognitive Discrepancy ($-S_{dis}$) causes a 3.7% accuracy drop, while removing Holistic Ambiguity ($-S_{ambi}$) results in a 1.9% drop. Using only the Thinking model's confidence ($C^T(q)$-only) degrades accuracy to 84.5%, demonstrating that the inter-model comparison is essential for reliable difficulty estimation.

Training data efficiency of DAMI-Pref. Figure 6 compares DAMI-Pred and DAMI-Pref under varying training data sizes. DAMI-Pref achieves near-optimal performance with only 1K samples and shows minimal improvement beyond, demonstrating strong robustness to limited supervision. In contrast, DAMI-Pred requires significantly more data to stabilize: it does not surpass the Thinking baseline until 4K samples. This gap confirms our motivation for preference-based estimation: pairwise comparisons provide more robust learning signals than pointwise regression, particularly in low-data

Question: The volume of the cylinder shown is $45\\pi$ cubic cm. What is the height in centimeters of the cylinder? [asy]\nsize(120); draw(shift(2.2,0)*yscale(0.3)*Circle((0,0), 1.2)); draw((1,0)--(1,-2));\ndraw((3.4,0)--(3.4,-2));\n\ndraw((1,-2)..(2.2,-2.36)..(3.4,-2));\n\nlabel(\"h\",midpoint((3.4,0)--(3.4,-2)),E);\n\ndraw (((2.2,0)--(3.4,0)));\n\nlabel(\"$r=3$\",midpoint((2.2,0)--(3.4,0)),N);\n\n[/asy]

Original Response

This is a complex or challenging question, and it is difficult to provide a direct and correct answer. I need to think about it. Well, so I need to find the height of this cylinder. The problem says the volume is 45π cubic cm. <omitted> V = πr²h. Solve for h: h = V / (πr²). Plug in V = 45π, r = 3. So h = (45π) / (π * 3²) = (45π) / (9π) = 45/9 = 5. Yep, the π cancels, 45 divided by 9 is 5. So height is 5 cm. Wait, is there any chance the diagram is tricking me? <omitted> Let me recap to make sure I didn't skip any steps.<omitted> </think> To find the **height** of a cylinder when given its **volume** and **radius**, <omitted> Final Answer: $$\n\\boxed{5}\n$$.

Token number: 2846

DEER

This is a complex or challenging question, and it is difficult to provide a direct and correct answer. I need to think about it. Well, so I need to find the height of this cylinder. The problem says the volume is 45π cubic cm. <omitted> V = πr²h. Solve for h: h = V / (πr²). Plug in V = 45π, r = 3. So h = (45π) / (π * 3²) = (45π) / (9π) = 45/9 = 5. Yep, the π cancels, 45 divided by 9 is 5. So height is 5 cm. Wait, (exit here) </think> To determine the height of the cylinder, we begin by recalling the standard formula for the volume of a cylinder: $$ V = \\pi r^2 h$$ Where: $ V $ is the volume, <omitted> Final Answer: $$\n\\boxed{5}\n$$.

Token number: 1188

LLM Routing

This is a complex or challenging question and it is difficult to provide a direct and correct answer. I need to think about it. Well, so I need to find the height of a cylinder when they give me the volume and some info about the radius. Let me start by recalling the formula for the volume of a cylinder. Hmm, volume is base area times height, right? And the base of a cylinder is <omitted> Wait, let me make sure I'm not misinterpreting the label. <omitted> Let me confirm once more. <omitted> Wait, hold on, sometimes people confuse radius and diameter in diagrams. <omitted> </think> To determine the height of the cylinder <omitted> Final Answer: $$\n\\boxed{5}\n$$.

Token number: 3017

DAMI's (λ=0.4)

We are given that the volume of the cylinder is $ 45\\pi $ cubic centimeters, and we are to find the height $ h $. The formula for the volume of a cylinder is:$$\nV = \\pi r^2 h$$ From the diagram, we are told that the radius $ r = 3 $ cm. Substitute the known values into the volume formula: $$45\\pi = \\pi (3)^2 $$ Simplify $$45\\pi = \\pi \\cdot 9 \\cdot h$$ Divide both sides by $ \\pi $:\n\n$$\n45 = 9h\n$$ Now solve for $ h $:\n$$ h = \\frac{45}{9} = 5\n$$ ### Final Answer:$$\\boxed{5} The height of the cylinder is $ \\boxed{5} $ centimeters</think> The volume of the cylinder is given as $45\\pi$ cubic cm, and the radius is $r = 3$ cm. <omitted> Final Answer: $$\n\\boxed{5}\n$$.

Token number: 351

Figure 8: Comparison of responses generated by four methods. DAMI produces efficient reasoning by modulating cognitive configuration.

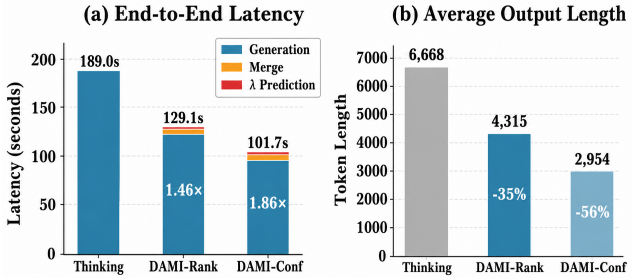

Figure 9: Computational cost analysis on MATH-500 (Qwen3-4B). (a) End-to-end latency breakdown showing λ prediction, parameter merging, and generation time. (b) Average output token length.

regimes where label noise has greater impact.

Case study. Figure 8 compares responses on a geometry problem from MATH-500. The Thinking model produces 2,846 tokens with typical overthinking patterns. DEER reduces output to 1,188 tokens through early termination, but cannot skip the lengthy content before the exit marker. This reflects a fundamental limitation of output control methods. LLM Routing, constrained to discrete model selection, routes this query to Thinking and yields 3,017 tokens. In contrast, DAMI generates only 351 tokens with qualitatively different behavior: direct problem formulation, concise calculation, and no redundant verification. This illustrates the core advantage of capability control: DAMI produces inherently efficient reasoning by modulating cognitive configuration.

Generalization on Multimodal Tasks. Beyond text-only models, we evaluate DAMI on the multimodal model Qwen3-VL-8B [Bai *et al.*, 2025]. We curate a diverse evaluation suite comprising 5 video understanding benchmarks (Video-Holmes, MMVU, VideoMMMU, VideoMME, LVBench) and 4 image-text reasoning benchmarks (WeMath, MathVista, MMMU, MathVision). As shown in Figure 7, DAMI-Conf consistently outperforms all baselines across both modalities. On video understanding tasks, DAMI-Conf achieves 64.0% average accuracy, surpassing the Thinking model by 2.5% absolute. On image-text reasoning tasks, DAMI-Conf attains 66.5%, a 1.6% improvement over Thinking. These findings confirm that our DAMI framework generalizes effectively to widely multimodal tasks. Moreover, it reveals that DAMI not only enables efficient reasoning, but also effectively integrates the complementary knowledge and reasoning capabilities of both models, leading to consistent performance gains across diverse benchmarks.

Computational cost. Figure 9 presents end-to-end latency. Despite introducing additional steps for λ prediction and parameter merging, both methods achieve substantial speedups over vanilla Thinking. The overhead from λ prediction is modest: DAMI-Pref only requires scoring each candidate λ with a lightweight reward model, while DAMI-Conf extracts confidence from short answer-only generations rather than full reasoning chains. Parameter merging adds $\sim$5.3s when loading models on-demand, but this can be reduced to under 1s by pre-loading both checkpoints into GPU memory. These overheads are negligible compared to generation time, which dominates total latency. By reducing output length by 35% (DAMI-Pref) and 56% (DAMI-Conf), our methods achieve 1.46$\times$ and 1.86$\times$ end-to-end speedups respectively.

5 Conclusion

This paper addresses the challenge that System 1's efficiency and System 2's reasoning capability cannot be readily unified within a single model. Unlike existing methods that constrain what models produce (output control), we propose capability control, which modulates how models think through dynamic parameter interpolation. Our DAMI framework offers two complementary methods, training based DAMI-Pref and training free DAMI-Conf, to achieve query-adaptive reasoning intensity. Beyond the technical contributions, this work suggests a broader perspective: the dichotomy between fast and slow thinking need not be a binary choice resolved at training time. The monotonicity and continuity properties we establish indicate that reasoning depth exists on a smooth spectrum, accessible through simple linear operations in parameter space.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62472419, 62472420) and the Enterprise Project (No. E4V06811F3).

Contribution Statement

Chenxu Yang and Qingyi Si contributed equally to this work. Zheng Lin is the corresponding author.

References

[Akhauri *et al.*, 2025] Yash Akhauri, Anthony Fei, Chi-Chih Chang, Ahmed F. AbouElhamayed, Yueying Li, and Mohamed S. Abdelfattah. Splitreason: Learning to offload reasoning, 2025.

[Arora and Zanette, 2025] Daman Arora and Andrea Zanette. Training language models to reason efficiently, 2025.

[Bai *et al.*, 2025] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[Cobbe *et al.*, 2021] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[Dai *et al.*, 2025] Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models, 2025.

[DeepSeek-AI *et al.*, 2025] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[Fan *et al.*, 2025] Siqi Fan, Bowen Qin, Peng Han, Shuo Shang, Yequan Wang, and Aixin Sun. The price of a second thought: On the evaluation of reasoning efficiency in large language models, 2025.

[Frankle *et al.*, 2020] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis, 2020.

[Han *et al.*, 2025] Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware LLM reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24842–24855, Vienna, Austria, July 2025. Association for Computational Linguistics.

[He *et al.*, 2025] Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning, 2025.

[Hendrycks *et al.*, 2021] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

[Hou *et al.*, 2025] Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning, 2025.

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

[Li *et al.*, 2025] Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhiwei Li, Bao-Long Bi, Ling-Rui Mei, Junfeng Fang, Xiao Liang, Zhijiang Guo, Le Song, and Cheng-Lin Liu. From system 1 to system 2: A survey of reasoning large language models, 2025.

[Luo *et al.*, 2025] Haotian Luo, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao, Dacheng Tao, and Li Shen. Ada-r1: Hybrid-cot via bi-level adaptive reasoning optimization, 2025.

[Ong *et al.*, 2025] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data, 2025.

[OpenAI, 2024] OpenAI. Learning to reason with llms, September 2024.

[Schumann *et al.*, 2023] Raphael Schumann, Elman Mansimov, Yi-An Lai, Nikolaos Pappas, Xibin Gao, and Yi Zhang. Backward compatibility during data updates by weight interpolation, 2023.

[Shen *et al.*, 2025] Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models, 2025.

[Shukor *et al.*, 2025] Mustafa Shukor, Louis Bethune, Dan Busbridge, David Grangier, Enrico Fini, Alaaeldin El-Nouby, and Pierre Ablin. Scaling laws for optimal data mixtures, 2025.

[Su and Cardie, 2025] Jinyan Su and Claire Cardie. Thinking fast and right: Balancing accuracy and reasoning length with adaptive rewards, 2025.

[Team, 2025] Qwen Team. Qwen3 technical report, 2025.

[Wang *et al.*, 2020] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging, 2020.

[Wang *et al.*, 2023] Lanrui Wang, Jiangnan Li, Chenxu Yang, Zheng Lin, and Weiping Wang. Enhancing empathetic and emotion support dialogue generation with prophetic commonsense inference. *CoRR*, abs/2311.15316, 2023.

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

[Wu *et al.*, 2025b] Taiqiang Wu, Runming Yang, Tao Liu, Jiahao Wang, and Ngai Wong. Revisiting model interpolation for efficient reasoning, 2025.

[Xia *et al.*, 2025] Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms, 2025.

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

[Yang *et al.*, 2024] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning, 2024.

[Yang *et al.*, 2025a] Chenxu Yang, Ruipeng Jia, Mingyu Zheng, Naibin Gu, Zheng Lin, Siyuan Chen, Weichong Yin, Hua Wu, and Weiping Wang. Weights-rotated preference optimization for large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26152–26175, Suzhou, China, November 2025. Association for Computational Linguistics.

[Yang *et al.*, 2025b] Chenxu Yang, Qingyi Si, Mz Dai, Dingyu Yao, Mingyu Zheng, Minghui Chen, Zheng Lin, and Weiping Wang. Test-time prompt intervention, 2025.

[Yang *et al.*, 2025c] Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models, 2025.

[Yao *et al.*, 2025] Yuxuan Yao, Shuqi Liu, Zehua Liu, Qintong Li, Mingyang Liu, Xiongwei Han, Zhijiang Guo, Han Wu, and Linqi Song. Activation-guided consensus merging for large language models, 2025.

[Zhan *et al.*, 2025] Keyao Zhan, Puheng Li, and Lei Wu. Analyzing the role of permutation invariance in linear mode connectivity, 2025.

[Zhang *et al.*, 2025] Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning, 2025.