MASKED SELF-DISTILLATION: INTERNALIZING THE CHAIN-OF-THOUGHT IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Reasoning Models (LRMs) produce long, explicit chains of intermediate steps before generating a final answer at inference time. These intermediate traces dominate latency, memory usage, and serving cost, even though the final answer correctness is not causally related to the trace correctness and the trace length is not a reliable indicator of the problem complexity. This raises an obvious question: can the computation expressed in these intermediate tokens be internalized into the parameters of a language model, enabling it to produce answers with much shorter intermediate traces? We propose **masked self-distillation**, a knowledge-distillation based post-training framework in which copies of the same model are instantiated as teacher and student, and the student model is trained to internalize all or part of the intermediate trace, thus becoming more efficient at inference. We vary the fraction of the intermediate trace the student is trained to internalize, interpolating between full internalization (the student emits only the solution) and no internalization (the student emits the full trace before the answer). We conduct controlled experiments on two reasoning domains: math and graph coloring. We use the masked self-distillation framework to post-train Qwen3-4B and Qwen3-8B models. Our results demonstrate that this method can be used to improve task performance while increasing inference efficiency across various domains and model sizes. We systematically analyze whether the improved efficiency gains in the post-trained models generalize to out-of-distribution (OOD) problems. We find that the masked self-distillation models generalize well for in-domain OOD problems, and the masked self-distillation training does not induce catastrophic forgetting in the student model on out-of-domain problems. Furthermore, our ablation study shows that supervised fine-tuning can train models to produce shorter traces, but at the cost of generalization, highlighting the importance of on-policy training in masked self-distillation. Lastly, we show that this framework can also be used effectively in a dual-model setting where a larger reasoning model teacher supervises a smaller non-reasoning student model, resulting in improved task performance and inference efficiency.

1 INTRODUCTION

Large Reasoning Models (LRMs), post-trained with RL, achieve strong performance on math, code, and planning benchmarks by generating long chains of intermediate tokens before producing a final answer. While these traces improve task performance, they also significantly increase inference cost. LRMs spend most of their generation budget on intermediate tokens rather than the answer, and inference latency, KV-cache footprint, and energy consumption all grow roughly linearly with trace length. At deployment, this is the main cost of serving them.

A broad line of work aims to reduce this cost for LRMs by adding trace length-control objectives during RL post-training [L1 Aggarwal & Welleck (2025), CoT-Valve Ma et al. (2025), O1-Pruner Luo et al. (2025), Kimi-1.5-style length-penalty rewards Team et al. (2025), GFPO Shrivastava et al. (2025), Arora & Zanette (2025)]. These methods reward shorter traces, however, the trace length compression is dependent on the base model's output distribution to generate a shorter trace during training. While the fact that performance increases with intermediate traces is well-known, the reasons for it are less clear. Recent work has shown that there is no causal connection between trace correctness and final-answer correctness (Valmeekam et al., 2025; Bhambri et al., 2025; Chen et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2025; Arcuschin et al., 2025), and the trace length is not connected to the computational complexity of the problem instance being solved (Shojaee* et al., 2025; Palod et al., 2025). These findings raise an obvious question: If the model's reasoning is not causally connected to its solution, is it necessary for the model to explicitly produce these tokens? Can the computation expressed in these intermediate tokens be internalized into the parameters of a language model?

Building on Hinton et al. (2015); Shenfeld et al. (2026); Hübotter et al. (2026), we study whether we can use knowledge distillation to internalize the information carried by intermediate tokens into the model's parameters. We propose **masked self-distillation**, a knowledge-distillation framework in which copies of the same model are instantiated as teacher and student, where the teacher is conditioned on the prompt and part of the reasoning trace, and the student is conditioned only on the prompt. The training objective is to minimize the reverse-KL divergence (Kullback & Leibler, 1951) between the teacher and student's probability distributions on responses generated by the student. This methodology enables the student to receive dense token-level feedback and leverage the benefits of RL on-policy exploration. Since the teacher is conditioned on a prefix of the reasoning trace, the student is incentivized to act as though the computations in that prefix were already in its context, which in turn leads it to produce shorter traces. We vary the percentage of trace provided to the teacher, thus controlling the amount of information the student has to internalize.

We instantiate this framework for training Qwen3-4B and Qwen3-8B (Yang et al., 2025) models across two reasoning domains: math and graph coloring. We first investigate whether we can use this framework to post-train models to internalize the information present in their reasoning tokens. In both domains and across different models, our results demonstrate that masked self-distillation improves task performance while reducing inference cost. We extensively investigate whether the efficiency gains generalize to out-of-distribution (OOD) problems. We conduct experiments in two OOD settings: (1) in-domain but outside the training distribution, and (2) out-of-domain - evaluating a model trained on graph coloring problems on math problems and vice versa. We find that the masked self-distillation models generalize well for in-domain OOD problems, and the masked self-distillation training does not induce catastrophic forgetting in the student model on out-of-domain problems. Furthermore, we perform an ablation study to examine the role of reverse-KL on-policy training vs. supervised fine-tuning (SFT) in token reduction. We find that while SFT improves in-distribution performance, the masked self-distillation variants consistently outperform the SFT variants on OOD test sets. This suggests that the reverse-KL on-policy training incentivizes the models to learn efficient reasoning behavior rather than just memorizing shorter traces. Finally, we also instantiate this framework in a dual-model setting where a larger reasoning model teacher supervises a different smaller non-thinking student model. Our studies show that the framework can also be effectively leveraged for training smaller models to improve task performance with efficient intermediate token generation.

2 Related Work

Test-time compute and Inference cost reduction: Wei et al. (2022); Kojima et al. (2023) showed that LLMs perform significantly better on planning and reasoning tasks when they are prompted to produce intermediate steps before generating the final answer. Since then a new class of models called Large Reasoning Models (LRMs) have emerged that are trained to generate reasoning traces before giving the final answer (Jaech et al., 2024; Comanici et al., 2025; Guo et al., 2025). While there has been a significant performance increase in planning and reasoning domains, LRMs spend most of their generation budget on intermediate tokens. A broad line of token efficiency work has emerged that aims to reduce the intermediate token length at inference. During inference, there have been various prompting methods proposed to incentivize efficient reasoning behavior in an LRM (Han et al., 2025; Zhao et al., 2026; Wang et al., 2025). While they have shown some success in reducing inference cost, prompting methods depend heavily on the steerability of the LRM, and their efficacy is highly task-dependent. To induce efficient reasoning behavior in the model, there have been RL post-training methods that aim to reduce the length of the traces (Aggarwal & Welleck, 2025; Team et al., 2025; Shrivastava et al., 2025; Arora & Zanette, 2025; Luo et al., 2025; Ma et al., 2025). The general theme of these works is that during RL training, they reward responses with shorter traces and correct solutions sampled from the base model. These works have shown promising reductions in trace lengths. However, the compression is dependent on the base model's output distribution to generate a shorter trace during training for RL to reward them. In contrast, during the masked

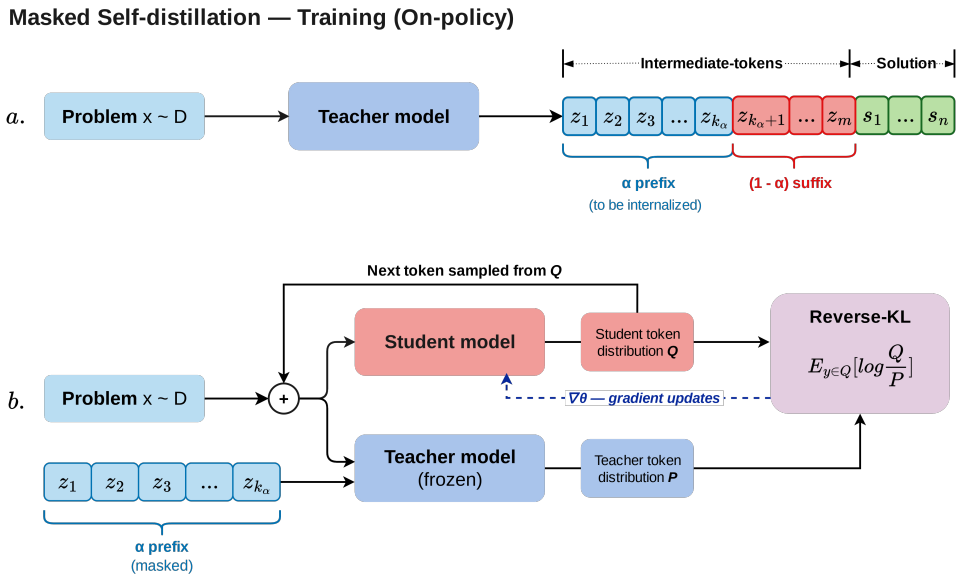


Figure 1: **Overview of the masked self-distillation framework.** *a.* For each problem $x \sim \mathcal{D}$, we sample a response from the teacher model. From each response, we extract the intermediate tokens z and the solution tokens s. *b.* The teacher π^T and the student π_θ^S are copies of the same model. The frozen teacher is conditioned on x and the first α fraction of z, while the student is conditioned only on x, so it has to internalize this α fraction into its parameters. The student's distribution Q is matched to the teacher's distribution P with the on-policy reverse-KL loss $\mathbb{E}_{y\sim Q}[\log Q/P]$, computed on student-sampled responses; only the student is updated.

self-distillation training, a portion of the reasoning trace is masked from the student, and the student is trained to internalize the information in the fraction of the reasoning trace. Thus, in our method, there is a supervision signal that incentivizes the production of shorter traces at inference.

Knowledge Distillation in LLMs: Knowledge distillation is widely studied to transfer information from a larger, more capable teacher model to a smaller student model (Hinton et al., 2015; Romero et al., 2014; Kim & Rush, 2016; Sanh et al., 2019). Snell et al. (2022); Gu et al. (2024); Agarwal et al. (2024) showed that this technique can be applied to LLM training as well. Snell et al. (2022) introduced the concept of context distillation in a setting similar to supervised learning: the idea of sampling from a model provided with extra context and training the same model to mimic these predictions without that context. There have been works that used self-distillation to internalize additional context such as factual information or environmental feedback into the model's parameters (Scheurer et al., 2023; Dou et al., 2024; Zhou et al., 2026; Kujanpää et al., 2024; Shenfeld et al., 2026; Hübotter et al., 2026). While these works explore the internalization of additional context, they do not study it for LRMs and reducing the intermediate token length.

CoT compression and Latent reasoning: Prior to the emergence of LRMs, some works investigated whether the Chain-of-Thought (CoT) tokens of an LLM can be internalized into the model's parameters (Snell et al., 2022; Deng et al., 2023; 2024). These methods use an off-policy training objective, and while the in-distribution accuracy can usually be matched with increased inference efficiency, they do not investigate whether these gains are generalizable. Recently, there is an additional line of work that studies reasoning in latent space (Hao et al., 2024; Saunshi et al., 2025; Su et al., 2025). These works suggest that the computations of the reasoning tokens can be compressed into the model parameters and thus, the models do not need to explicitly generate these tokens.

3 Masked Self-Distillation

We present a knowledge-distillation framework, which we call *masked self-distillation*, in which two copies of the same model are instantiated as a teacher and a student. The teacher copy of the model operates in think mode, while the student copy is initialized in no-think mode. The initial part (or

all) of the trace is *masked* from the student during the distillation i.e., the teacher sees it, but the student does not and the student model is trained to internalize the masked portion into its parameters. The higher the amount of trace the student internalizes, the shorter its outputs and the cheaper its inference, because that computation has moved into the student's parameters during distillation.

Figure 1 shows an overview of the *masked self-distillation* framework. First, for each problem $x \in \mathcal{D}$, we sample a response from the teacher and take the segment between the <*think*> and </*think*> tags as the intermediate tokens $z = (z_1, \ldots, z_m)$, where m is the length of the reasoning trace. The simplest way to do masked distillation on the teacher response is to train the student with SFT on the suffix part of the intermediate tokens followed by the solution tokens. However, as SFT is an off-policy method, it is known to suffer from exposure bias (Pomerleau, 1991; Ross & Bagnell, 2010) and does not generalize well (Chu et al., 2025; Shenfeld et al., 2026). We therefore train the student using on-policy distillation. The student is only conditioned on x and is trained to match the teacher's next-token distribution, conditioned on x and a fractional prefix of z. We control what fraction of the trace is masked with a parameter $\alpha \in [0, 1]$. The teacher is conditioned on x and the first $k_\alpha = \lfloor \alpha\, m \rfloor$ tokens of the trace, $z_{\leq k_\alpha}$, while the student is conditioned on x alone. Since the teacher is now effectively going to produce only the tokens after the α prefix, the student effectively has to internalize the α prefix of the trace.

Let $y = (y_1, \ldots, y_L)$ be a response sampled from the student, where L is the length of the response. The objective is to minimize the reverse-KL divergence between the student and teacher next-token distributions, computed on student-sampled responses:

$$\mathcal{L}^{\alpha}_{\mathrm{MSD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi^S_\theta(\cdot|x)} \left[\sum_{t=1}^{L} D_{\mathrm{KL}}\big(\pi^S_\theta(\,\cdot \mid x, y_{<t}) \,\big\|\, \pi^T(\,\cdot \mid x, z_{\leq k_\alpha}, y_{<t})\big) \right], \tag{1}$$

where, for next-token distributions p and q over the vocabulary $\mathcal{V}$,

$$D_{\mathrm{KL}}(q \parallel p) = \mathbb{E}_{v\sim q}\left[\log \frac{q(v)}{p(v)}\right] = \sum_{v\in\mathcal{V}} q(v) \log \frac{q(v)}{p(v)}. \tag{2}$$

Because y is sampled from the student, the objective is on-policy: the student learns from the responses it actually generates, and it receives dense token-level feedback from the teacher at every position instead of a single sparse reward.

As $\alpha \in [0, 1]$, masked self-distillation has two extreme cases. In the first case ($\alpha = 1$ thus $k_\alpha = m$), the teacher is conditioned on question x and the full reasoning trace, while the student is only conditioned on x. The student has to internalize the entire reasoning process and, at inference, emit only the solution tokens. We use the term *fully masked* to denote this training. In the second case ($\alpha = 0$, $k_\alpha = 0$), the teacher and the student are both conditioned only on x. We use the term *non-masked* to refer to this training.

4 Experimental Setup

Our broad aim is to study how the masked self-distillation training affects inference cost and task performance. We conduct experiments in the following domains -

1. **Math -** We used the popular *GSM8K* as our in-distribution dataset. This dataset, introduced by (Cobbe et al., 2021), consists of grade school math problems, and is designed to evaluate the reasoning capabilities of large language models. It contains 8.5K problems, each paired with a question and an answer. The dataset is divided into 7.5K training problems and 1K test problems. To test the out-of-distribution (OOD) generalization of models post-trained on GSM8K problems, we have in-domain OOD and out-of-domain test sets. We use AIME-25 (on Hugging Face, 2025) and MATH-500 (Lightman et al., 2024) as in-domain OOD test sets as these datasets share the same arithmetic substrate but contain problems of higher difficulty. For the out-of-domain test, we use the graph coloring $\mathrm{GC\text{-}ER}_{\chi_{3-4}}$ test set (described below).
2. **Graph Coloring -** Graph coloring is a canonical NP-complete reasoning problem that is related to both constraint satisfiability as well as practical problems like scheduling and

allocation. In this domain, the model is given a graph, specified by vertices and edges, and the number of possible colors and the task is to return a configuration where no two connected vertices have the same color. For a graph G, the minimum number of colors required to properly color its vertices is the chromatic number, denoted $\chi(G)$. We construct a custom dataset using the `networkx` Python library (Hagberg et al., 2008), controlling problem difficulty by computing $\chi(G)$ for each sampled graph and retaining only those with the desired chromatic number. Our training set consists of 10,000 problems on Erdős–Rényi graphs (ERDdS & R&wi, 1959) with $\chi(G) \in \{3, 4\}$. The in-distribution test set consists of 500 held-out problems. We use the notation $\text{GC-ER}_{\chi_{3-4}}$ to denote this set. For evaluating generalization, we have in-domain OOD and out-of-domain test sets. For in-domain generalization, we have test sets with harder problems GC-ER_{χ_6} and GC-BA_{χ_6} generated using the Erdős–Rényi and Barabási–Albert algorithms (Barabási & Albert, 1999) respectively (refer to appendix B for more details). For the out-of-domain test, we use the *GSM8K* test set.

Training details: Our implementation builds on the `verl` library (Sheng et al., 2024). We train Qwen3-4B and Qwen3-8B models (Yang et al., 2025) and perform α masked self-distillation post-training in math and graph coloring domains. First, we generate responses from the teacher (base model in think mode) on the training data and then keep only the problems that the teacher solved correctly, i.e., rejection sampling (Figure 1a). We perform the masked self-distillation training (Figure 1b) on the student (base model in no-think mode) on the dataset created after rejection sampling. We vary $\alpha \in \{0.0, 0.3, 0.5, 0.7, 1.0\}$ to control how much of the trace the student internalizes and to study how this affects its accuracy and trace length. We provide additional hyperparameter details in the appendix section A.

5 Results and Discussion

5.1 Effect of Internalization on Task performance and inference efficiency

In this section, we study whether models post-trained using the masked self-distillation framework internalize the information present in the teacher's reasoning tokens and achieve better task performance with improved inference efficiency.

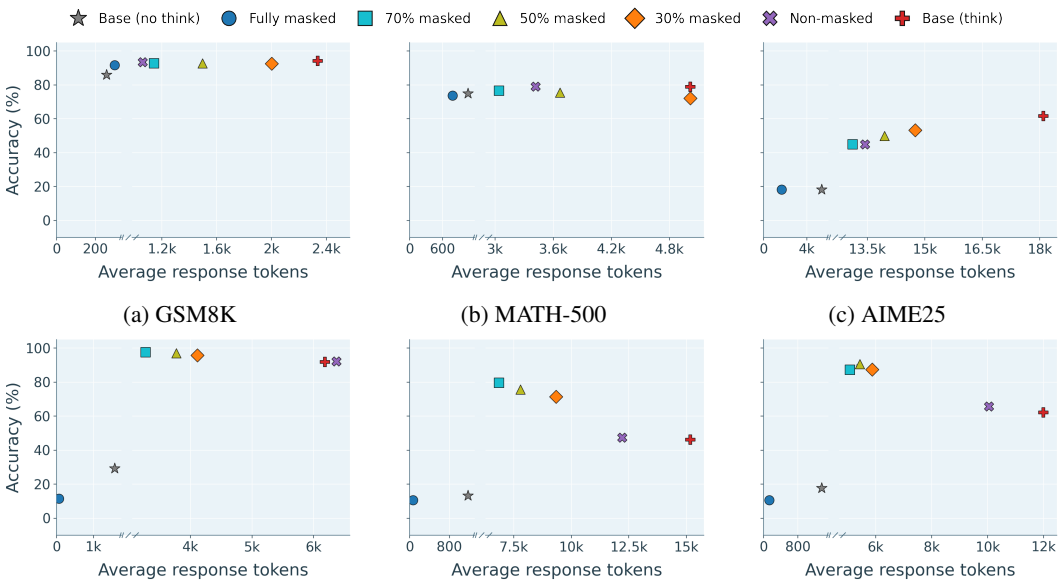
Figure 2 shows accuracy vs. average response length for *masked self-distillation* across different values of α, together with the base student and teacher models, averaged over 3 random seeds [1], on the in-distribution and out-of-distribution datasets for the math and graph coloring domains for the Qwen3-4B model. Figure 3 shows the same comparison for the Qwen3-8B model.

As shown in Figure 2a, the student (base no-think) accuracy is 86% on the GSM8K test set while producing ∼260 tokens on average, while the teacher's (base think) accuracy is 94% and it produces ∼2.3k tokens on average. After we train the student models on the GSM8K training set with masked self-distillation for different values of α, they produce far fewer tokens than the teacher while keeping similar performance. The fully masked variant ($\alpha = 1$) gives the best trade-off: it reaches ∼92% accuracy with about $8\times$ fewer tokens than the teacher.

On the graph coloring in-distribution test set (Figure 2d), the student model accuracy is 29%, and it generates ∼1.5k tokens on average, while the teacher reaches 92% with ∼6k tokens. After training with masked self-distillation, the student produces shorter outputs, and the variants with $\alpha \in \{0.3, 0.5, 0.7\}$ exceed the teacher by 4–5 points, while producing 1.5-2x fewer tokens. Unlike on GSM8K, the fully masked training is not effective here. The fully masked model produces only ∼70 tokens, but its accuracy drops to 12%, even less than the base student model. This suggests that fully internalizing the reasoning process is not feasible for the graph-coloring domain, and the student still requires a scaffold of intermediate tokens to perform well.

We observe two patterns in the average response length across different values of α. (1) In the math domain, the trace generated by the α-masked variants becomes shorter as α increases, with one exception: the non-masked variant ($\alpha = 0$) produces about as many tokens as the $\alpha = 0.7$ variant. (2) In the graph coloring domain, the variants with $\alpha \in \{0.3, 0.5, 0.7\}$ form a cluster and produce nearly the same number of tokens, within a few hundred tokens of each other, while the non-masked variant

[1] We report the confidence intervals in appendix C.2 and C.3.



(a) GSM8K (b) MATH-500 (c) AIME25

(d) GC-ER$_{\chi_{3-4}}$ (e) GC-ER$_{\chi_6}$ (f) GC-BA$_{\chi_6}$

Figure 2: *Qwen3-4B Pareto plot*: Accuracy vs. mean response length comparison on math (top row) and graph-coloring (bottom row) domain test-datasets for α-masked variants, teacher (base think) and student (base no-think). Note: Accuracy and mean response are averaged over 3 random seeds.

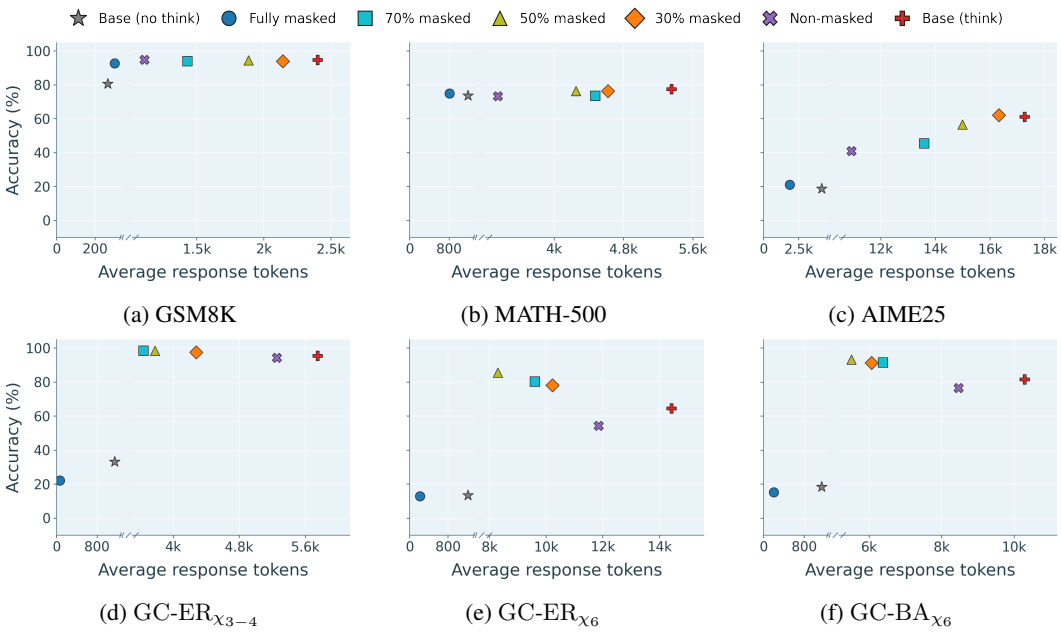
produces the most tokens among the trained variants. We attribute this behavior to the mode-seeking nature of the training objective (Gu et al., 2024; Agarwal et al., 2024). Since we train the student with on-policy distillation, minimizing the reverse-KL divergence between the student's and the teacher's token distributions, the objective strongly penalizes the student for placing probability mass where the teacher's probability mass is small or zero. The student therefore concentrates on the high-density regions (modes) of the teacher and ignores its low-probability tails. The teacher can have several such modes, and which one the student converges to depends on the trajectories it samples during training. In masked self-distillation, the student is trained to behave as though a fraction of the reasoning trace were already in its context, and it tends to generate the remaining part of the trace, but this may not correspond to the exact trace the teacher would have generated. If the student happens to generate shorter or longer traces during training, and the teacher assigns high probability to them, the student can converge to a teacher mode that favors traces of that length. We believe this is what happens in both domains: for $\alpha = 0$ in the math domain, the student converges to a mode that produces traces almost as short as those of the $\alpha = 0.7$ variant, while for $\alpha = 0$ in the graph coloring domain, it converges to a mode that produces even more tokens than the teacher on the in-distribution test set.

The Qwen3-8B model shows the same pattern in both domains (Figures 3a and 3d). On GSM8K, the fully masked variant reaches 93% accuracy, close to the teacher's 95%, with about $8\times$ fewer tokens. On graph coloring, the variants with $\alpha \in \{0.3, 0.5, 0.7\}$ match the teacher's accuracy of 95% while producing about 1.4–$1.6\times$ fewer tokens.

While it is extremely difficult to do a systematic study of the content of the reasoning traces expressed in natural language, our qualitative analysis results suggest that the traces produced by the α-masked variants seem disconnected, as if beginning from an intermediate point in the reasoning (refer to appendix sections F and E for more details). Taken together, these results suggest that the student models indeed learn to internalize the computations present in the α fraction of the teacher's trace and thus become more efficient at inference.

5.2 Analyzing the out-of-distribution effectiveness

We want to evaluate whether the masked self-distillation training leads to memorization of shorter traces on in-distribution (ID) problems at the expense of generalization. We extensively study the



(a) GSM8K (b) MATH-500 (c) AIME25

(d) GC-ER$_{\chi_{3-4}}$ (e) GC-ER$_{\chi_6}$ (f) GC-BA$_{\chi_6}$

Figure 3: *Qwen3-8B Pareto plot*: Accuracy vs. mean response length comparison on math (top-row) and graph-coloring (bottom-row) domain test-datasets for α-masked variants, teacher (base think) and student (base no-think). Note: Accuracy and mean response are averaged over 3 random seeds.

out-of-distribution (OOD) performance in two OOD settings: (1) in-domain but outside the training distribution, and (2) out-of-domain.

In-domain OOD: On the math-domain OOD datasets, MATH-500 and AIME-25 (Figures 2b and 2c), we observe the same trend as on GSM8K. On MATH-500, all α-masked variants match the teacher's accuracy while producing far fewer tokens. On AIME-25, the $\alpha = 0.3$ variant reaches about 55% accuracy, close to the teacher, while generating about $1.3\times$ fewer tokens. The remaining α-masked variants are less accurate than the teacher but are more inference-efficient.

The graph coloring OOD results are extremely promising (Figures 2e and 2f). The teacher reaches only 45% accuracy on GC-ER$_{\chi_6}$ and 62% on GC-BA$_{\chi_6}$, while the best α-masked variant ($\alpha = 0.7$) reaches 80% on GC-ER$_{\chi_6}$ and 89% on GC-BA$_{\chi_6}$. This is $\sim$30 percentage points above the teacher, with about half as many tokens. We attribute this to the on-policy nature of masked self-distillation: the student is trained explicitly on this domain, on trajectories generated by its own policy.

The in-domain OOD results for Qwen3-8B are consistent with those for Qwen3-4B (Figure 3b–3c and Figure 3e–3f) on both domains. Taken together, these results indicate that a student trained with masked self-distillation generalizes robustly on in-domain OOD problems, matching the teacher's performance in the math domain and surpassing it in the graph coloring domain, while being more inference-efficient.

Out-of-domain Generalization: Since we train the student models on a single domain (math or graph coloring), a natural question is whether the resulting gains in both accuracy and inference cost carry over to other domains, or whether the models overspecialize to the training domain and lose accuracy elsewhere. To test this, we perform out-of-domain evaluation tests.

Figure 4 shows out-of-domain evaluation analysis for α-masked Qwen3-8B models, together with the base student and teacher models. The models trained on the graph coloring domain retain their inference cost savings even on the GSM8K test dataset. However, the models trained on the GSM8K dataset have slightly lower performance on GC-ER$_{\chi_{3-4}}$ test set compared to the base teacher model. On the GC-ER$_{\chi_{3-4}}$ test set, the teacher model reaches 95.5% accuracy with $\sim$5.7k tokens, while the best masked self-distillation variant ($\alpha = 0.3$) reaches 87% with $\sim$4.6k tokens, and $\alpha = 0.5$ reaches 85.5% with $\sim$4.2k tokens. These variants remain far above the base student model (33%), so the training still improves performance on this harder task, but the token cost reduction is minimal. These results show that while the inference cost gains are domain-dependent, the masked self-distillation

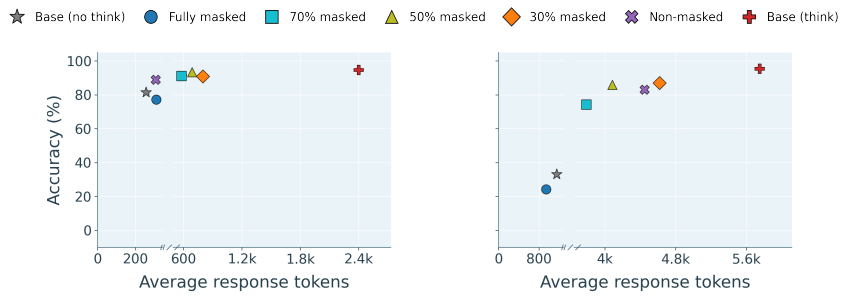


Figure 4: *Out-of-domain Qwen3-8B Pareto plots:* (left) graph coloring models evaluated on the GSM8K test set and (right) GSM8K models evaluated on GC-ER$_{\chi_{3-4}}$ test set.

training does not induce catastrophic forgetting in the student models. We report the out-of-domain analysis for α-masked Qwen3-4B model in Appendix figure 8.

5.3 Effect of the maximum token budget

The Pareto plots report a single operating point per baseline, namely its accuracy and mean token length [2] under a fixed cap. To see the full trade-off, we also plot the accuracy curve over the token budget: for each budget b on the x-axis, the curve reports the accuracy the model would achieve if generation were stopped at b tokens. A problem counts as solved only if the model produces a correct answer within b tokens. Figure 5 shows these curves for all baselines on the GC-ER$_{\chi_{3-4}}$ test set under an 8k cap for Qwen3-4B model. Masked self-distillation shifts the entire curve to the left. The curves for $\alpha \in \{0.3, 0.5, 0.7\}$ lie above and to the left of the teacher's over the whole range. This demonstrates that the α-masked variants perform better than the base teacher at every fixed token budget. Notably, at a budget of 4k tokens, the $\alpha = 0.3$ variant already solves about 80% of the problems, while the teacher solves fewer than 50%.

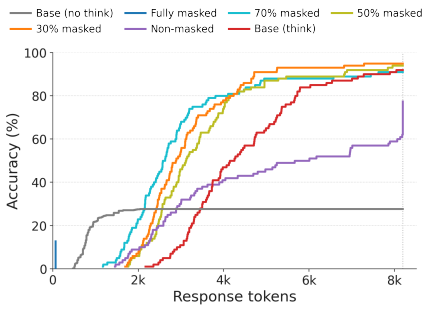


Figure 5: *Qwen3-4B*: Accuracy curve over max-token budget.

5.4 Ablation Study: SFT vs on-policy distillation

We compare the α-masked variants trained with SFT and masked self-distillation to examine whether the effectiveness of our framework comes from the internalization of the teacher's reasoning traces together with the student's on-policy exploration for shorter reasoning traces, or whether similar gains can be achieved simply through over-training using SFT on a fraction of the teacher's reasoning trace and the final solution. We provide the details of the SFT training procedure in the appendix A.3.

Table 1 reports the accuracy and mean response length of the SFT and masked self-distillation trained α-masked variants. We post-train the Qwen3-4B model in no-think mode on variants of the graph-coloring training set where each variant has question, an α fraction of the trace, and the solution. The models are evaluated on the graph-coloring in-domain OOD test sets and on the out-of-domain GSM8K test set. From the table, we observe that for $\alpha \in \{0.3, 0.5, 0.7\}$, the variants trained with the on-policy reverse-KL objective generalize better to both the in-domain OOD and out-of-domain test sets than their SFT counterparts. For the fully masked variant ($\alpha = 1$), SFT appears to overfit to the training distribution: although it performs well on the in-distribution test set, its performance drops substantially on the OOD test sets. Moreover, on the out-of-domain test set, the SFT-trained fully masked variant collapses, whereas the reverse-KL-trained fully masked variant largely maintains the

[2]The mean response length does not provide a complete picture of the response-length distribution over the test set. To examine how masked distillation changes the response-length distribution and what distributions the α-masked variants follow, we report violin plots in Appendix D.

performance of the base model. Furthermore, across all values of α, the SFT-trained variants generate substantially longer responses on the cross-domain test set than their reverse-KL counterparts. This suggests that simply training on shortened teacher traces through SFT is not sufficient for the student to consistently learn shorter reasoning behavior. In contrast, reverse-KL training combines the internalization of the teacher's reasoning trace with on-policy trajectories generated by the student itself, leading to shorter responses while better preserving generalization.

	GC-ER$_{\chi_{3-4}}$		GC-ER$_{\chi_6}$		GC-BA$_{\chi_6}$		GSM8K	
Masked	SFT	Reverse KL	SFT	Reverse KL	SFT	Reverse KL	SFT	Reverse KL
100%	**88.7**	11.5	**11.4**	10.5	**26.2**	10.6	0.0	**87.0**
	68	67	68	66	68	115	32k*	299
70%	94.2	**97.5**	52.7	**79.5**	72.1	**88.3**	68.5	**87.7**
	2631	3274	5899	6906	4681	5185	4866	549
50%	95.1	**97.0**	60.2	**75.5**	82.3	**89.9**	77.0	**90.8**
	3431	3772	7816	7986	5631	5410	9084	1051
30%	94.1	**95.8**	58.5	**71.1**	82.7	**87.3**	82.2	**89.6**
	4211	4112	10299	9384	6661	5942	5501	933
0%	**95.4**	92.3	**55.9**	46.7	**70.1**	65.7	84.8	**92.6**
	5433	6369	13369	12439	10248	10186	4606	1670

Table 1: *Qwen3-4B variants* : Accuracy and mean token length comparison between SFT vs. on-policy trained α-masked variants. Each cell reports accuracy (%) on the first line and mean token length on the second. Bold marks the highest accuracy; underline marks the shortest mean response. "*" denotes the model collapses and exhausts its token generation budget.

5.5 Ablation Study: Dual Model Setting

In our *masked self-distillation* framework, we use copies of the same model as student and teacher, where the student is in no-think mode and teacher in think mode. But our framework can be instantiated in dual model setting as well, in which a larger reasoning model operates as teacher and supervises a smaller non-thinking student model. We evaluate the *dual-model* setting on graph coloring, where teacher is Qwen3-8B (in think mode) and student is Qwen3-1.7B (in no-think mode).

Figure 7 in the appendix shows accuracy vs. mean token length for *dual-model* setting across different values of α, together with the base student model, teacher model, and the base model in reasoning mode on the in-distribution and out-of-distribution datasets. We observe a trend similar to that in the self-distillation setting. The α-masked variants, $\alpha \in \{0.3, 0.5, 0.7\}$, internalize a fraction of the teacher's reasoning traces and generate substantially fewer tokens on average compared to the base reasoning model while achieving accuracies that are 30-40 percentage points higher on the in-distribution and both out-of-distribution test datasets. These results demonstrate that the *masked distillation* framework can also be used to post-train smaller models to reason more efficiently.

6 Conclusion

In this work, we study whether the computation expressed in the intermediate tokens can be internalized into the parameters of a language model. We proposed *masked self-distillation*, a knowledge-distillation based post-training method that can be utilized to induce token-efficient reasoning without any external feedback or supervision. Our results across math and graph coloring demonstrate that masked self-distillation consistently improves task performance while reducing intermediate tokens. We also conducted experiments in the countdown domain and find similar results (refer to Appendix C.1). We extensively studied the generalization effectiveness of the masked self-distillation training. Our findings demonstrate that models post-trained using the framework generalize well on in-domain OOD problems and do not suffer from catastrophic forgetting on out-of-domain problems. We also compare with SFT trained models and show how the on-policy reverse-KL objective trains the models to generate shorter intermediate traces rather than just memorizing the shorter trace completions. Finally, we show that the masked distillation training can be effectively leveraged for training smaller models to improve task performance with efficient intermediate token generation.

AI USE STATEMENT

We used OpenAI's ChatGPT models to assist with minor paraphrasing and language refinement during the editing process. All technical content and original writing are our own, and no text was generated by the tool beyond rewording.

REPRODUCIBILITY STATEMENT

We have provided all the information regarding our training setup in section 4, datasets in section 4 and appendix B, and hyperparameters used in appendix A that can be used for replicating our experiments. We will release all the code datasets and models publicly in the camera-ready version.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. Chain-of-thought reasoning in the wild is not always faithful. *arXiv preprint arXiv:2503.08679*, 2025.

Daman Arora and Andrea Zanette. Training language models to reason efficiently, 2025. *URL https://arxiv. org/abs/2502.04463*, 2025.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286 (5439):509–512, 1999.

Siddhant Bhambri, Upasana Biswas, and Subbarao Kambhampati. Interpretable traces, unexpected outcomes: Investigating the disconnect in trace-based knowledge distillation. *arXiv preprint arXiv:2505.13792*, 2025.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*, 2025.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

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. *arXiv preprint arXiv:2110.14168*, 2021.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Countdown. Countdown (game show). `https://en.wikipedia.org/wiki/Countdown_(game_show)`. [Accessed 13-05-2025].

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*, 2024.

Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. Re-rest: Reflection-reinforced self-training for language agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15394–15411, 2024.

PARW ERDdS and A R&wi. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

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

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman (eds.), *Proceedings of the 7th Python in Science Conference*, pp. 11 – 15, Pasadena, CA USA, 2008.

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 (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24842–24855, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1274. URL https://aclanthology.org/2025.findings-acl.1274/.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

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

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL https://arxiv.org/abs/2205.11916.

Kalle Kujanpää, Pekka Marttinen, Harri Valpola, and Alexander Ilin. Efficient knowledge injection in llms via self-distillation. *arXiv preprint arXiv:2412.14964*, 2024.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 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)*, pp. 6025–6035, 2025.

Contributors on Hugging Face. American invitational mathematics examination 2025 (aime25) dataset, 2025. URL https://huggingface.co/datasets/math-ai/aime25.

Vardhan Palod, Karthik Valmeekam, Kaya Stechly, and Subbarao Kambhampati. Performative thinking? the brittle correlation between cot length and problem complexity. *arXiv preprint arXiv:2509.07339*, 2025.

Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *CoRR*, abs/1412.6550, 2014. URL https://api.semanticscholar.org/CorpusID:2723173.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 661–668. JMLR Workshop and Conference Proceedings, 2010.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations*, volume 2025, pp. 14855–14881, 2025.

Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. Training language models with language feedback at scale. *arXiv preprint arXiv:2303.16755*, 2023.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Parshin Shojaee*, Iman Mirzadeh*, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. In *NeurIPS*, 2025. URL https://arxiv.org/abs/2506.06941.

Vaishnavi Shrivastava, Ahmed Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample more to think less: Group filtered policy optimization for concise reasoning. *arXiv preprint arXiv:2508.09726*, 2025.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

DiJia Su, Hanlin Zhu, Yingchen Xu, Jiantao Jiao, Yuandong Tian, and Qinqing Zheng. Token assorted: Mixing latent and text tokens for improved language model reasoning. *arXiv preprint arXiv:2502.03275*, 2025.

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

Karthik Valmeekam, Kaya Stechly, Vardhan Palod, Atharva Gundawar, and Subbarao Kambhampati. Beyond semantics: The unreasonable effectiveness of reasonless intermediate tokens. *arXiv preprint arXiv:2505.13775*, 2025.

Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. Wait, we don't need to "wait"! removing thinking tokens improves reasoning efficiency. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 7459–7482, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.394. URL https://aclanthology.org/2025.findings-emnlp.394/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903, 2022. URL https://arxiv.org/abs/2201.11903.

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

Mengjiao Sherry Yang, Dale Schuurmans, Pieter Abbeel, and Ofir Nachum. Chain of thought imitation with procedure cloning. *Advances in Neural Information Processing Systems*, 35:36366–36381, 2022.

Guangxiang Zhao, Qilong Shi, Xusen Xiao, Xiangzheng Zhang, Tong Yang, and Lin Sun. Thinking with reasoning skills: Fewer tokens, more accuracy. In Yunyao Li, Georg Rehm, and Mei Tu (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 2295–2308, San Diego, California, USA, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-394-4. doi: 10.18653/v1/2026.acl-industry.154. URL https://aclanthology.org/2026.acl-industry.154/.

Ruiyang Zhou, Shuozhe Li, Amy Zhang, and Liu Leqi. Expo: Unlocking hard reasoning with self-explanation-guided reinforcement learning. *Advances in Neural Information Processing Systems*, 38:90989–91016, 2026.

CONTENTS

1 Introduction **1**

2 Related Work **2**

3 Masked Self-Distillation **3**

4 Experimental Setup **4**

5 Results and Discussion **5**

5.1 Effect of Internalization on Task performance and inference efficiency 5

5.2 Analyzing the out-of-distribution effectiveness 6

5.3 Effect of the maximum token budget 8

5.4 Ablation Study: SFT vs on-policy distillation 8

5.5 Ablation Study: Dual Model Setting 9

6 Conclusion **9**

A Hyperparameter details **16**

A.1 Masked Self-Distillation 16

A.2 Dual-model setting 16

A.3 SFT 16

B Dataset Details **16**

C Additional Results **17**

C.1 Countdown Results 17

C.2 Qwen3-4B Results 17

C.3 Qwen3-8B Results 18

C.4 Dual-model setting Results 19

C.5 Out-of-Domain Qwen3-4B Results 19

D Response length distributions **20**

E Prompt format and the responses of masked self distilled models **21**

F Qualitative analysis of masked-self distillation variants' traces **25**

A HYPERPARAMETER DETAILS

A.1 MASKED SELF-DISTILLATION

In our masked-self-distillation experiments, the student uses a maximum prompt length of $1{,}024$ tokens and a maximum response length of $8{,}192$ tokens. The teacher uses a longer context length of $10{,}240$ tokens to accommodate the fraction of intermediate tokens (ITs), collected in Phase 1 along with the question. Each training step uses a question batch size of 16. The total number of training steps is 200. We observe that the models are trained to convergence. For the 4B models, we use a mini-batch size of 1 and 8 rollouts per question. For the 8B models, we use 5 rollouts per question. The mini-batch size remains 1 for GSM8K and is increased to 2 for the 8B graph-coloring experiments. We optimize with AdamW at a constant learning rate of 1×10^{-6}, with no warmup and weight decay 0.01. For all runs, we clip the actor gradient L2 norm at 1. Rollouts at training time are sampled with the vLLM inference engine at temperature 1.0, top-$p = 1.0$, and top-$k = -1$. All experiments are run on a single node using between 2 and 8 Intel Gaudi2 HPUs, depending on the model size and task.

A.2 DUAL-MODEL SETTING

The dual-model experiments use Qwen3-1.7B as the student and a frozen Qwen3-8B teacher. We use a question batch size of 16, a mini-batch size of 1, and 5 rollouts per question. The optimizer, training steps, sequence lengths, rollout sampling parameters, and gradient clipping follow the masked self-distillation setting above. All dual-model setting experiments are run on a single node with 4 Intel Gaudi2 HPUs.

A.3 SFT

We additionally study a supervised fine-tuning (SFT) variant of masked distillation, in which the student is trained on teacher collected responses (Figure 1a) using the standard cross-entropy loss. For teacher generated responses, let z be the intermediate trace and s the solution tokens that follow it. The SFT target for a given α is the suffix of the trace followed by the solution, $y^\alpha = (z_{k_\alpha+1}, \ldots, z_m, s)$, and the student is conditioned only on x:

$$\mathcal{L}^{\alpha}_{\mathrm{SFT}}(\theta) = -\,\mathbb{E}_{x\sim\mathcal{D},\,(z,s)\sim\pi^T(\cdot|x)}\left[\sum_{t=1}^{|y^\alpha|}\log \pi^S_\theta(y^\alpha_t \mid x, y^\alpha_{<t})\right]. \tag{3}$$

For our graph-coloring α-masked SFT experiments, we fine-tune Qwen3-4B using a global batch size of 16 and a micro-batch size of 1. We use 30k training steps. We optimize with AdamW at a learning rate of 1×10^{-5}, and no weight decay. We use a linear warmup (10 steps) followed by cosine decay to zero, and clip the gradient L2 norm at 1. The maximum sequence length is $10{,}240$ tokens, with a maximum prompt length of $1{,}024$ tokens and a maximum response length of $8{,}192$ tokens. All SFT experiments are run on a single node with 4 Intel Gaudi2 HPUs.

Note: During inference we keep the model temperature 0.6, top-$p = 0.95$, top-$k = 20$, and max response length $= 32k$.

B DATASET DETAILS

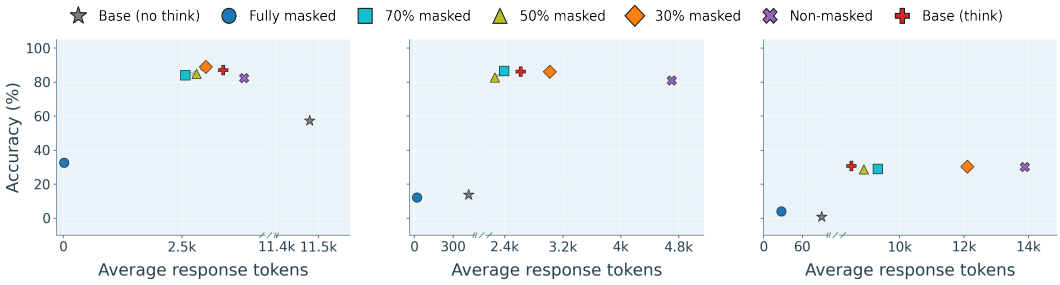
Graph Coloring: We generate graph instances using two standard random graph algorithms to produce structurally distinct topologies. The Erdős–Rényi (ER) algorithm (ERDdS & R&wi, 1959) constructs a graph on n vertices by including each of the $\binom{n}{2}$ possible edges independently with probability p. The resulting degree distribution is binomial and concentrates sharply around its mean $p(n-1)$, so ER graphs are approximately homogeneous: most vertices have a similar number of neighbors. For our in-distribution datasets, we fix the number of vertices to $n = 10$ and edges to $e = 18$ to control even the density of the graph. For generating harder instances in the GC-ER$_{\chi_6}$, we increase the density of the graph by increasing number of edges to $e = 30$ and filter the generated instances to have a chromatic number of 6.

The Barabási–Albert (BA) algorithm (Barabási & Albert, 1999) generates graphs through preferential attachment. Starting from a small seed graph, vertices are added one at a time, and each new vertex attaches m edges to existing vertices with probability proportional to their current degree. This algorithm generates heterogeneous graphs with a small number of high-degree hubs. In the GC-BA_{χ_6} test set, we have fixed the number of vertices to $n = 10$ and edges to $e = 30$ and filter the generated instances to have a chromatic number of 6. All graphs are generated using NetworkX (Hagberg et al., 2008).

Countdown: This domain (Countdown.) is a generalized version of the classic 24 Game (Yang et al., 2022), where the objective is to combine a set of input numbers using basic arithmetic operations (+, –, ×, ÷) to reach a specified target number. In our ID dataset, each problem consists of 3 to 4 two-digit input numbers, with the target number also being a two-digit number. The dataset contains 8.5K examples, split into 8K training instances and 500 test instances. For OOD testing, we have three datasets, each perturbing a different axis. *Target-range shift* keeps the input cardinality and digit count fixed but shifts the target to a one-digit number (500 problems). *Search-depth shift* keeps the digit counts fixed but increases the input cardinality to 5 numbers respectively (500 problems).

C Additional Results

C.1 Countdown Results



(a) ID (b) Target-range shift (c) Search-depth shift

Figure 6: *Qwen3-4B Pareto plot*: accuracy vs. mean response length comparison on countdown domain test datasets for α-masked variants, teacher (base think) and student (base no-think).

Figure 6 shows the Pareto plot for Qwen3-4B on the countdown test sets for all baselines. The α-masked variants ($\alpha \in \{0.3, 0.5, 0.7\}$) reach the same level of accuracy as the teacher on every countdown test set. On the in-distribution and target-shift sets, however, they save only a few hundred tokens relative to the teacher, and on the search-depth-shift set they produce more tokens than the teacher. This may be due to the mode-seeking behavior of the reverse-KL objective discussed in Section 5.1.

C.2 Qwen3-4B Results

Tables 2 and 3 report the accuracy and mean response length for all α-masked variants, together with the base student and teacher models, on the math and graph-coloring test datasets, respectively.

	Masked fraction					Base	
Dataset	100%	70%	50%	30%	0%	think	no-think
GSM8K	$91.4_{\pm 0.5}$	$92.9_{\pm 0.4}$	$92.8_{\pm 0.2}$	$92.7_{\pm 0.5}$	$93.5_{\pm 0.5}$	$\mathbf{94.2}_{\pm 0.1}$	$86.1_{\pm 0.4}$
	$302_{\pm 3}$	$1124_{\pm 12}$	$1536_{\pm 71}$	$1961_{\pm 78}$	$1061_{\pm 79}$	$2332_{\pm 27}$	$\underline{258}_{\pm 2}$
MATH500	$74.1_{\pm 1.0}$	$75.9_{\pm 1.0}$	$75.7_{\pm 0.1}$	$74.9_{\pm 1.1}$	$\mathbf{78.9}_{\pm 0.1}$	$76.9_{\pm 0.3}$	$75.2_{\pm 0.7}$
	$\underline{763}_{\pm 29}$	$3080_{\pm 125}$	$3671_{\pm 61}$	$4722_{\pm 557}$	$3340_{\pm 146}$	$5137_{\pm 214}$	$1051_{\pm 34}$
AIME25	$17.8_{\pm 1.9}$	$44.4_{\pm 1.9}$	$51.1_{\pm 1.9}$	$54.4_{\pm 6.9}$	$46.7_{\pm 3.3}$	$\mathbf{61.1}_{\pm 5.1}$	$18.9_{\pm 1.9}$
	$\underline{1609}_{\pm 116}$	$13146_{\pm 1500}$	$13730_{\pm 1206}$	$14673_{\pm 154}$	$12598_{\pm 2081}$	$18190_{\pm 840}$	$5552_{\pm 1560}$

Table 2: *Qwen3-4B, math domain*: accuracy and mean token length for each masked fraction, trained on GSM8K. Each entry is the mean over 3 random seeds, with the sample standard deviation in the subscript. Accuracy (%) is on the first line; the smaller line is mean response length in tokens. Bold marks the highest accuracy in each row; underline marks the shortest mean response.

	Masked fraction					Base	
Dataset	100%	70%	50%	30%	0%	think	no-think
GC-ER$_{\chi_{3\text{-}4}}$	$11.5_{\pm 0.3}$	$\mathbf{97.5}_{\pm 1.0}$	$97.0_{\pm 0.7}$	$95.8_{\pm 1.1}$	$92.3_{\pm 0.6}$	$91.9_{\pm 0.6}$	$29.2_{\pm 1.4}$
	$\underline{67}_{\pm 0}$	$3274_{\pm 129}$	$3772_{\pm 115}$	$4112_{\pm 129}$	$6369_{\pm 120}$	$6185_{\pm 237}$	$1576_{\pm 85}$
GC-ER$_{\chi_6}$	$10.5_{\pm 0.2}$	$\mathbf{79.5}_{\pm 1.5}$	$75.5_{\pm 0.7}$	$71.1_{\pm 1.5}$	$46.7_{\pm 1.6}$	$45.4_{\pm 2.5}$	$13.7_{\pm 2.1}$
	$\underline{66}_{\pm 3}$	$6906_{\pm 196}$	$7986_{\pm 389}$	$9384_{\pm 344}$	$12439_{\pm 487}$	$15489_{\pm 643}$	$1254_{\pm 166}$
GC-BA$_{\chi_6}$	$10.6_{\pm 0.3}$	$88.3_{\pm 1.0}$	$\mathbf{89.9}_{\pm 1.3}$	$87.3_{\pm 1.9}$	$65.7_{\pm 1.8}$	$62.1_{\pm 2.6}$	$18.4_{\pm 1.9}$
	$\underline{115}_{\pm 36}$	$5185_{\pm 135}$	$5410_{\pm 152}$	$5942_{\pm 382}$	$10186_{\pm 366}$	$12134_{\pm 298}$	$1566_{\pm 445}$

Table 3: *Qwen3-4B, graph coloring domain*: accuracy and mean token length for each masked fraction, trained on GC-ER$_{\chi_{3\text{-}4}}$. Each entry is the mean over 3 random seeds, with the sample standard deviation in the subscript. Accuracy (%) is on the first line; the smaller line is mean response length in tokens. Bold marks the highest accuracy in each row; underline marks the shortest mean response.

C.3 QWEN3-8B RESULTS

Tables 4 and 5 report the accuracy and mean response length for all α-masked variants, together with the base student and teacher models, on the math and graph-coloring test datasets, respectively.

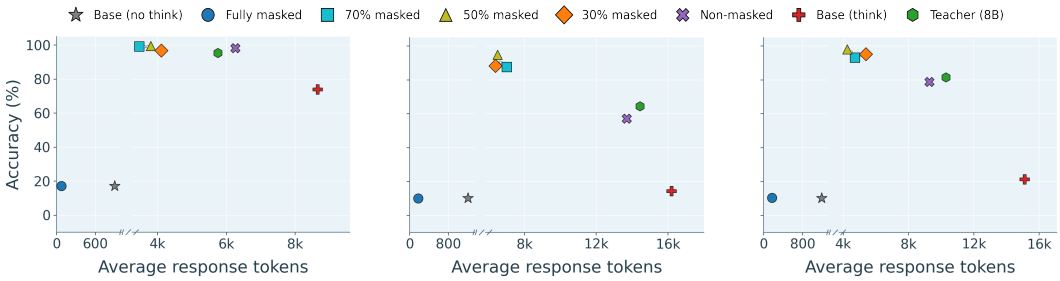
	Masked fraction					Base	
Dataset	100%	70%	50%	30%	0%	think	no-think
GSM8K	$92.8_{\pm 0.1}$	$94.0_{\pm 0.2}$	$94.5_{\pm 0.1}$	$93.9_{\pm 0.5}$	$\mathbf{95.0}_{\pm 0.3}$	$94.6_{\pm 0.2}$	$80.7_{\pm 0.4}$
	$302_{\pm 1}$	$1429_{\pm 30}$	$1884_{\pm 45}$	$2141_{\pm 91}$	$1109_{\pm 12}$	$2402_{\pm 6}$	$\underline{266}_{\pm 2}$
MATH500	$74.9_{\pm 0.2}$	$73.5_{\pm 0.8}$	$76.3_{\pm 0.4}$	$76.3_{\pm 0.1}$	$73.4_{\pm 1.4}$	$\mathbf{77.5}_{\pm 0.1}$	$73.8_{\pm 2.5}$
	$\underline{805}_{\pm 37}$	$4472_{\pm 365}$	$4249_{\pm 77}$	$4619_{\pm 42}$	$3348_{\pm 64}$	$5355_{\pm 116}$	$1181_{\pm 120}$
AIME25	$21.1_{\pm 1.9}$	$45.6_{\pm 8.4}$	$56.7_{\pm 3.3}$	$\mathbf{62.2}_{\pm 3.8}$	$41.1_{\pm 13.5}$	$61.1_{\pm 1.9}$	$18.9_{\pm 1.9}$
	$\underline{1848}_{\pm 561}$	$13587_{\pm 746}$	$14989_{\pm 1333}$	$16322_{\pm 297}$	$10924_{\pm 1271}$	$17274_{\pm 752}$	$4121_{\pm 1077}$

Table 4: *Qwen3-8B, math domain*: accuracy and mean token length for each masked fraction, trained on GSM8K. Each entry is the mean over 3 random seeds, with the sample standard deviation in the subscript. Accuracy (%) is on the first line; the smaller line is mean response length in tokens. Bold marks the highest accuracy in each row; underline marks the shortest mean response.

	Masked fraction					Base	
Dataset	100%	70%	50%	30%	0%	think	no-think
GC-ER$_{\chi_{3\text{-}4}}$	$22.1_{\pm 0.5}$	**98.5**$_{\pm 0.6}$	**98.5**$_{\pm 0.8}$	$97.6_{\pm 0.5}$	$94.3_{\pm 1.0}$	$95.5_{\pm 0.3}$	$33.2_{\pm 1.8}$
	$\underline{62}_{\pm 0}$	$3650_{\pm 123}$	$3766_{\pm 233}$	$4277_{\pm 196}$	$5253_{\pm 69}$	$5752_{\pm 131}$	$1149_{\pm 28}$
GC-ER$_{\chi_6}$	$13.0_{\pm 0.1}$	$80.4_{\pm 1.0}$	**85.6**$_{\pm 0.6}$	$78.3_{\pm 2.2}$	$54.4_{\pm 1.9}$	$64.4_{\pm 2.5}$	$13.5_{\pm 0.5}$
	$\underline{211}_{\pm 25}$	$9598_{\pm 405}$	$8289_{\pm 181}$	$10224_{\pm 691}$	$11855_{\pm 266}$	$14434_{\pm 12}$	$1217_{\pm 23}$
GC-BA$_{\chi_6}$	$15.2_{\pm 0.9}$	$91.4_{\pm 0.9}$	**93.3**$_{\pm 1.0}$	$91.3_{\pm 0.4}$	$76.5_{\pm 1.9}$	$81.5_{\pm 0.4}$	$18.4_{\pm 1.9}$
	$\underline{198}_{\pm 27}$	$6304_{\pm 227}$	$5509_{\pm 164}$	$6148_{\pm 241}$	$8470_{\pm 290}$	$10298_{\pm 288}$	$1145_{\pm 22}$

Table 5: *Qwen3-8B, graph coloring domain*: accuracy and mean token length for each masked fraction, trained on GC-ER$_{\chi_{3\text{-}4}}$. Each entry is the mean over 3 random seeds, with the sample standard deviation in the subscript. Accuracy (%) is on the first line; the smaller line is mean response length in tokens. Bold marks the highest accuracy in each row; underline marks the shortest mean response.

C.4 Dual-model setting Results



(a) GC-ER$_{\chi_{3-4}}$ (b) GC-ER$_{\chi_6}$ (c) GC-BA$_{\chi_6}$

Figure 7: *Dual-model setting Pareto plot*: accuracy vs. average response length comparison on graph-coloring domain test datasets for α-masked variants, base think, student (base no-think) and teacher (8B think).

Figure 7 in the appendix shows accuracy vs. mean token length for *dual-model* setting across different values of α, together with the base student model, teacher model, and the base model in reasoning mode on the in-distribution and out-of-distribution datasets.

C.5 Out-of-Domain Qwen3-4B Results

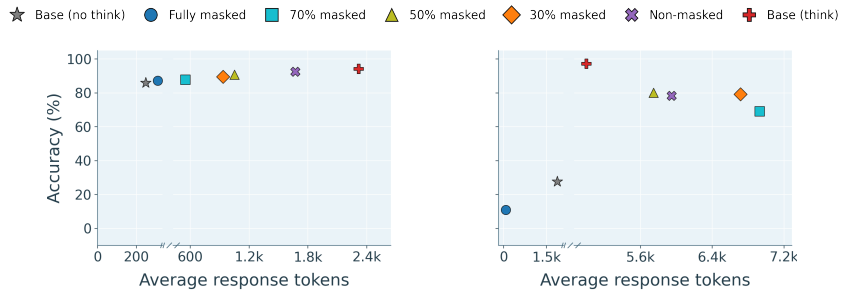


Figure 8: *Out-of-domain Qwen3-4B Pareto plots:* (left) graph coloring models evaluated on the GSM8K test set and (right) GSM8K models evaluated on GC-ER$_{\chi_{3-4}}$ test set.

Figure 8 shows out-of-domain evaluation for Qwen3-4B models, together with the base student and teacher model. The results are pretty similar to the Qwen3-8B out-of-domain results. The models

trained on the graph coloring domain retain their inference cost savings even on the GSM8K test dataset. However, the models trained on the GSM8K dataset have slightly lower performance on the GC-ER$_{\chi_{3-4}}$ test set compared to the base teacher model and also produce more tokens.

D RESPONSE LENGTH DISTRIBUTIONS

In the Pareto plots, to compare the inference efficiency of each baseline, we use the mean response length. However, the mean alone does not provide the full picture of how the model's response lengths are distributed across the entire test set. A high mean response length can arise either because the model consistently generates long responses across the test set or because it generates very long responses for only a small fraction of problems. In other words, the response-length distribution can exhibit either unimodal or bimodal behavior. To examine this, we plot violin plots, which combine a box plot with a kernel density estimate and therefore show the overall shape of the response-length distribution.

Figure 9 shows the violin plots for all baselines on the graph-coloring OOD-BA test set under a maximum token budget of 32k. From the figure, we can see that the teacher and the non-masked variant exhibit clear bimodal response-length distributions and produce responses with a median length of approximately 10k tokens. In contrast, the α-masked variants ($\alpha \in 0.3, 0.5, 0.7$) exhibit a largely uni-modal distribution with a median response length of approximately 3k tokens. This suggests that, by internalizing a fraction of the reasoning trace, the masked variants learn to generate shorter responses more consistently across the test set. Figures 10 and 11 show the violin plots for all domain test datasets for Qwen3-4B and Qwen3-8B respectively.

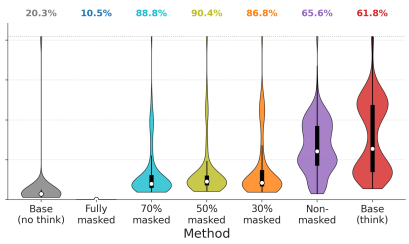
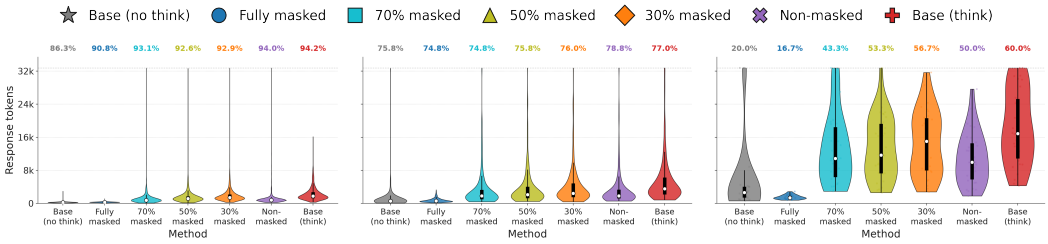


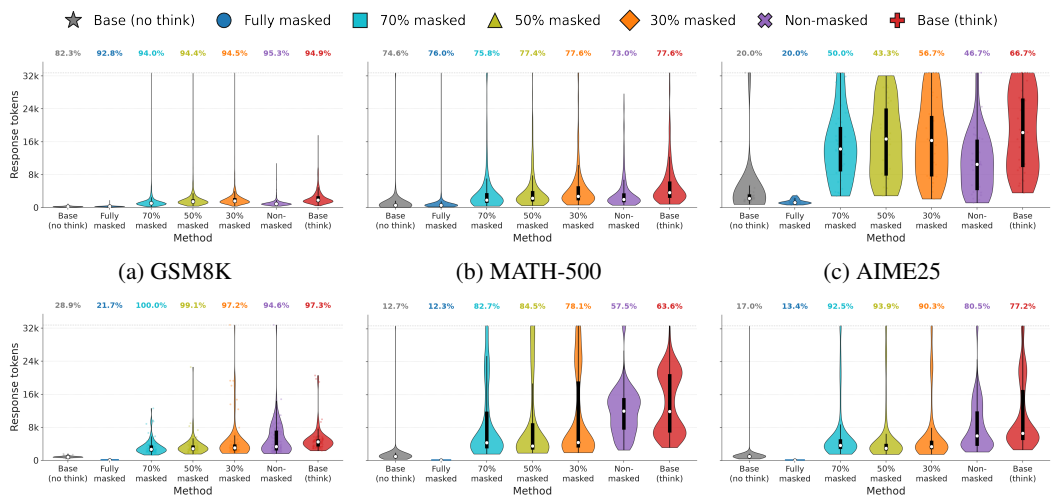
Figure 9: Violin Plot



(a) GSM8K (b) MATH-500 (c) AIME25

(d) GC-ER$_{\chi_{3-4}}$ (e) GC-ER$_{\chi_6}$ (f) GC-BA$_{\chi_6}$

Figure 10: *Qwen3-4B Violin plot*: Token distribution comparison on math (top row), graph-coloring (middle row) and countdown (bottom row) domain test datasets for α-masked variants, teacher (base think) and student (base no-think).



(a) GSM8K (b) MATH-500 (c) AIME25

(d) GC-ER$_{\chi_{3-4}}$ (e) GC-ER$_{\chi_6}$ (f) GC-BA$_{\chi_6}$

Figure 11: *Qwen3-8B Violin plot*: Token distribution comparison on math (top row), graph-coloring (middle row) and countdown (bottom row) domain test datasets for α-masked variants, teacher (base think) and student (base no-think).

E PROMPT FORMAT AND THE RESPONSES OF MASKED SELF DISTILLED MODELS

We show a GC-ER$_{\chi_{3-4}}$ test instance prompt and responses of every masked self-distilled variant. We can see that the non-masked variant produces a response that is very similar to the teacher. However, the responses of models trained with $\alpha \in \{0.3, 0.5, 0.7\}$ seem disconnected, as if beginning from an intermediate point in the reasoning rather than from the start.

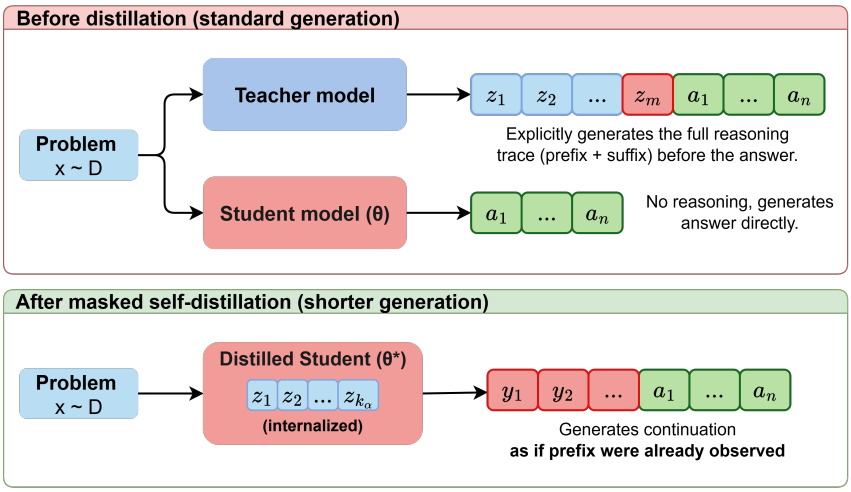


Figure 12: Illustration of standard generation before distillation and shorter generation after masked self-distillation.

Prompt

```
Please provide a coloring for this graph such that every vertex is not connected to a
vertex of the same color. The graph has these properties:

Vertices: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
Edges: [(0, 1), (0, 7), (0, 9), (1, 2), (1, 3), (1, 5), (1, 7), (1, 8), (1, 9), (2, 3),
(2, 4), (2, 8), (3, 4), (3, 6), (3, 8), (3, 9), (5, 6), (5, 9)]
Possible colors: [1, 2, 3, 4]

Return your final answer in <answer> </answer> tags as a JSON map of vertices to colors,
 for example <answer> {"0": 1, "1": 2, "2": 3} </answer>.
assistant
```

Qwen3-8B base, think (Teacher) response

```
<think>
Okay, let's see. I need to color this graph with four colors so that no two connected
vertices share the same color. The vertices are numbered 0 through 9, and there are a
bunch of edges listed. Let me start by understanding the graph structure.

First, I should probably draw the graph or at least note down which vertices are
connected to which. Let me list out the edges again:

Edges:
(0,1), (0,7), (0,9),
(1,2), (1,3), (1,5), (1,7), (1,8), (1,9),
(2,3), (2,4), (2,8),
(3,4), (3,6), (3,8), (3,9),
(5,6), (5,9)

So vertex 0 is connected to 1,7,9. Vertex 1 is connected to 0,2,3,5,7,8,9. That's a lot.
 Vertex 2 is connected to 1,3,4,8. Vertex 3 is connected to 1,2,4,6,8,9. Vertex 4 is
conn

[... 7421 characters of reasoning elided ...]

ct. Vertex 2 is color 1, neighbors are 1 (2), 3 (3), 4 (2), 8 (4). Correct. Vertex 7 is
color 3, neighbors are 0 (1), 1 (2). Correct. Vertex 1 is color 2, neighbors are 0 (1),
2 (1), 3 (3), 5 (1), 7 (3), 8 (4), 9 (4). All different. Vertex 0 is color 1, neighbors
are 1 (2), 7 (3), 9 (4). All different.

Yes, this seems to work. So the answer should be this coloring.
</think>

<answer>
{
  "0": 1,
  "1": 2,
  "2": 1,
  "3": 3,
  "4": 2,
  "5": 1,
  "6": 2,
  "7": 3,
  "8": 4,
  "9": 4
}
</answer>
```

Non-Masked variant response

```
<think>
Okay, let's tackle this graph coloring problem. The goal is to assign colors from
[1,2,3,4] to each vertex such that no two connected vertices share the same color. The
graph has vertices 0-9 and specific edges. Let me first try to understand the structure
of the graph.

First, I'll list out all the edges again to make sure I have them right:

Edges:
(0,1), (0,7), (0,9),
(1,2), (1,3), (1,5), (1,7), (1,8), (1,9),
(2,3), (2,4), (2,8),
(3,4), (3,6), (3,8), (3,9),
(5,6), (5,9)

So, each vertex is connected to others as per these edges. Let me try to visualize or
note the connections for each vertex.

Starting with vertex 0: connected to 1,7,9.
Vertex 1 is connected to 0,2,3,5,7,8,9. That
```

```
[... 4027 characters of reasoning elided ...]

), 1 (2), 3 (3), 5 (1). All different.

So this seems to work. Let me check if any conflicts were missed. For example, vertex 3
is connected to 9 (color 4) - yes, 3 is color 3. Vertex 5 is connected to 9 (color 4) -
5 is color 1. All okay.

So the color assignments are:

0:1, 1:2, 2:1, 3:3, 4:2, 5:1, 6:2, 7:3, 8:4, 9:4.

Which matches the correct solution provided. So that's the answer.
</think>

<answer>
{"0": 1, "1": 2, "2": 1, "3": 3, "4": 2, "5": 1, "6": 2, "7": 3, "8": 4, "9": 4}
</answer>
```

30%-Masked variant response

```
 1, 3, 5, 7, 8, 9. Wait, vertex 1 is connected to 0,2,3,5,7,8,9. So vertex 1's neighbors
 are 0,2,3,5,7,8,9. So vertex 1 is connected to 0,2,3,5,7,8,9. So vertex 1 is connected
to 0, which is color 1. So vertex 1's color can't be 1. Wait, but vertex 1's color is 1?
 Wait, no, I'm trying to assign colors. Let me think again.

Wait, maybe I need to start with vertex 0, then vertex 1, etc. Let me try to go step by
step.

Let me try to color vertex 0 first. Let's assign color 1 to vertex 0. Then, vertices
connected to 0 (1,7,9) can't be color 1.

Next, vertex 1. It's connected to 0 (color 1), so it can't be color 1. Let's assign
color 2 to vertex 1. Then, its neighbors (0,2,3,5,7,8,9) can't be col

[... 5654 characters of reasoning elided ...]

3 and 2 -> different. Edge (3,6): 3 and 2 -> different. Edge (3,8): 3 and 4 ->
different. Edge (3,9): 3 and 4 -> different. Edge (2,3): 1 and 3 -> different. Edge
(2,4): 1 and 2 -> different. Edge (2,8): 1 and 4 -> different. Edge (5,6): 1 and 2 ->
different. Edge (5,9): 1 and 4 -> different. All edges are okay.

So this seems to be a valid coloring. Therefore, the answer should be this mapping.
</think>

<answer>
{"0": 1, "1": 2, "2": 1, "3": 3, "4": 2, "5": 1, "6": 2, "7": 3, "8": 4, "9": 4}
</answer>
```

50%-Masked variant response

```
 1, 2, 3, 4, 5, 6, 7, 8, 9. Wait, vertex 1 is color 2, vertex 3 is color 3, vertex 5 is
color 1? Wait, no. Wait, vertex 5 is connected to 1 (color 1?), no. Wait, vertex 1 is
color 2. Wait, I need to track all the colors.

Wait, let me retrace. Let me list all vertices with their current colors:

Vertex 0: 1

Vertex 1: 2

Vertex 2: 1

Vertex 3: 3

Vertex 4: 2

Vertex 5: ?

Wait, I think I need to go step by step again. Maybe I made a mistake in the order.

Let me try again, more carefully.

Start with vertex 0. Assign color 1.

Vertex 0: 1

Now, vertex 1 is connected to 0. So vertex 1 can't be 1. Assign color 2.
```

```
Vertex 1: 2

Now, vertex 2 is connected to 1. So vertex 2 can't be 2. Assign col

[... 3053 characters of reasoning elided ...]

color 2, connected to 3 (3) and 5 (1). Correct.

So the final answer would be:

{
  "0": 1,
  "1": 2,
  "2": 1,
  "3": 3,
  "4": 2,
  "5": 1,
  "6": 2,
  "7": 3,
  "8": 4,
  "9": 4
}

But let me check if there's another possible coloring. For example, maybe some colors
could be swapped. But the question just requires any valid coloring. So this should be
acceptable.
</think>

<answer>
{"0": 1, "1": 2, "2": 1, "3": 3, "4": 2, "5": 1, "6": 2, "7": 3, "8": 4, "9": 4}
</answer>
```

70%-Masked variant response

```
 3, 3, 2, 4? Wait, vertex 3 is color 3, vertex 2 is color 1, vertex 4 is color 2, vertex
 5 is color 1, vertex 6 is color 1, vertex 7 is color 3, vertex 8 is color 4, vertex 9
is color 4. Wait, vertex 9 is connected to vertex 3 (color 3), vertex 5 (color 1),
vertex 0 (color 1), vertex 1 (color 1). So vertex 9 is color 4, which is different from
all its neighbors. So that's okay.

Wait, but vertex 9 is connected to vertex 3 (color 3), which is okay. Vertex 9 is color
4. So that's okay.

Wait, but vertex 9 is connected to vertex 1 (color 1), vertex 0 (color 1), vertex 3
(color 3), vertex 5 (color 1). So vertex 9's color is 4, which is different. So that's
okay.

So, the color assignments so far

[... 2806 characters of reasoning elided ...]

But this seems to work. Let me just write the JSON as per the example.

The vertices are 0-9. So the JSON would be:

{"0": 1, "1": 2, "2": 1, "3": 3, "4": 2, "5": 1, "6": 2, "7": 3, "8": 4, "9": 4}

But let me check again for vertex 6. Vertex 6 is color 2. It's connected to 3 (color 3)
and 5 (color 1). So okay.

Yes. So this should be correct.
</think>

<answer>
{"0": 1, "1": 2, "2": 1, "3": 3, "4": 2, "5": 1, "6": 2, "7": 3, "8": 4, "9": 4}
</answer>
```

Fully Masked variant response

```
{"0": 1, "1": 2, "2": 1, "3": 3, "4": 2, "5": 1, "6": 2, "7": 3, "8": 4, "9": 4}
```

F QUALITATIVE ANALYSIS OF MASKED-SELF DISTILLATION VARIANTS' TRACES

Prompt

System:

```
You are checking whether a model's generated reasoning trace reads as a coherent,
self-contained chain of thought, or whether it reads as if it is continuing a thought
that was never stated e.g. it opens using a connective ("So", "Then", "Thus",
"Given that", "Therefore"), a pronoun or definite reference with no antecedent ("it",
"this value", "the result", "that number").

You are judging ONLY internal coherence of the trace as a standalone piece of text.
Do not judge correctness of the reasoning or the final answer.
```

User:

```
Question given to the model: {question}

Model's generated trace (this is the ENTIRE generation, starting from token 1):
{reasoning_trace}

Answer these:
1. abrupt_start (true/false): does the trace's opening sentence read as though
   picking up mid-reasoning, rather than beginning to address the question?
2. dangling_references (true/false): does the opening 1-2 sentences
   contain a pronoun or definite-reference term (e.g. "it", "this",
   "that value", "the remaining number") with no antecedent stated
    earlier in the trace?
3. coherence_score (1-5): 1 = reads as a truncated fragment / picks up
   mid-thought, 5 = fully self-contained, well-formed opening.

Output JSON only:
{"abrupt_start": true/false, "dangling_references": true/false,
"coherence_score": 1-5}
```

Figure 13: Prompt used for the LLM-judge coherence audit. The judge is shown only the question and the trace

We perform a qualitative analysis of traces generated by the Qwen3-8B model and its α-masked variants on the $\text{GC-ER}_{\chi_{3-4}}$ test set. To check whether the student-generated traces start abruptly, as if continuing a partially done calculation, we perform an LLM-as-a-judge analysis where we provide the question and the reasoning trace to a coherence audit judge. The judge examines the trace and returns the following: (1) Abrupt start: whether the opening sentence reads as a continuation of an unstated thought rather than a self-contained start. (2) Coherence score: Measuring the coherence of the entire trace. (3) Dangling reference: whether in the opening sentences of the trace, there are reference terms that have no prior antecedent within the trace. We report the coherence audit judge prompt in Figure 13.

We use GPT-5-Mini in low reasoning mode as our coherence audit judge. We report our results in Table 6. We see that the audit judge correctly scores the teacher-generated responses and 0% masked model responses as coherent, with no abrupt starts and dangling references. However, the judge consistently flags the traces generated by the α masked traces as abrupt, incoherent, and having missing references. These preliminary qualitative results suggest that the student models internalize the computations present in the α fraction of the teacher's trace and learn to generate a shorter trace.

Metric	Base (think)	0% masked	30% masked	50% masked	70% masked
Abrupt-start rate	0.00	0.00	0.94	1.00	1.00
Mean coherence score (1–5)	5.00	5.00	2.09	1.42	1.29
Dangling-reference rate	0.00	0.00	0.24	0.91	0.90

Table 6: LLM-judge coherence analysis of traces generated by the Qwen3-8B model (teacher) and its α masked variants trained on the graph coloring ID test set