Masked Distillation: Internalizing the Chain-of-Thought in Language Models

Durgesh Kalwar*
dkalwar@asu.edu
SCAI, Arizona State University

Vardhan Palod*
vpalod@asu.edu
SCAI, Arizona State University

Subbarao Kambhampati
rao@asu.edu
SCAI, Arizona State University

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 a natural question: can the computation expressed in these intermediate tokens be internalized into the parameters of a language model, enabling it to produce answers directly (or with much shorter intermediate traces)? We introduce *masked distillation*, a knowledge-distillation framework in which a student LLM is trained to predict only the solution tokens conditioned on the question, while a reasoning teacher provides feedback on the student's responses after conditioning on the question and its own CoT trace. We instantiate this framework in two settings: (i) a *self-distillation* setting, in which the same model serves as the teacher in thinking mode and as the student in non-thinking mode, and (ii) a *dual-model* setting, in which a larger reasoning teacher supervises a separate smaller non-thinking student over the solution tokens. By treating intermediate tokens as a scaffold which reasoning models use to fit over the solution tokens, We additionally vary the length of intermediate-token scaffolding the student is supervised on, interpolating between full internalization (the student emits only the solution) and no internalization (the student emits the full trace before the answer). We evaluate the framework through controlled experiments on two reasoning domains: GSM8K (grade-school arithmetic) and Countdown (a number-puzzle search task). Our results show that the success of full internalization is sharply task-dependent and tracks the student's prior exposure to the task during pretraining: it works on GSM8K but fails on Countdown without inference-time scaffolding. However, providing the student models with a small scaffold closes the gap to self-distillation on Countdown. The task performance increases from 41.7% (fully masked) to 86.2%, essentially matching the teacher (87.3%) at $\sim 1.3\times$ fewer inference tokens than the non-masked variant.

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 scale, this is the main cost of serving them.

A broad line of work aims to reduce this cost for LRMs by adding 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, but the model still produces an explicit trace at inference, and the savings are bounded by how short a trace can be. While the fact of the performance increase with intermediate traces is well-known, the reasons for it are less clear.

*Equal contribution

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

Our recent work has shown that there is no causal connection between trace correctness and final-answer correctness (Valmeekam et al., 2025; Kambhampati et al., 2025; Bhambri et al., 2025), and the trace length is not connected to the computational complexity of the problem instance being solved (Palod et al., 2025). These findings raise the question that if model's reasoning is not causally connected to its solution, is it necessary for the model to explicitly produce these tokens and can the computation expressed in these intermediate tokens be internalized into the parameters of a language model?

A recent line of work related to self-distillation has shown that language models can be trained to internalize additional context using knowledge distillation (Hübotter et al., 2026; Shenfeld et al., 2026). The general idea of distilling from a larger, more capable teacher into a smaller student is itself well established (Gu et al., 2024; Agarwal et al., 2024); (Hübotter et al., 2026) and (Shenfeld et al., 2026) take this further by showing that even when the teacher and the student are the same model, the teacher's access to additional context (expert demonstrations or environment feedback) can be transferred into the student's parameters. Prior to the emergence of LRMs, there were works which investigated whether Chain-of-Thought (CoT) tokens could be internalized into a model's parameters. (Snell et al., 2022) introduced context distillation and showed that a T5-small can be distilled from a CoT-prompted version of itself into a direct-answer model that solves arithmetic problems. (Deng et al., 2023) generalized this with implicit chain-of-thought via knowledge distillation (ICoT-KD), and (Deng et al., 2024) later proposed stepwise internalization (ICoT-SI), which removes intermediate tokens stage by stage so that the teacher's reasoning is gradually compressed into the student's hidden states. A common pattern across these methods is that in-distribution accuracy can usually be matched increased inference efficiency, provided the training setup is right. However, the trade-off between extra training cost and inference savings is not well-studied. More importantly, whether the resulting students generalize is also an open question.

Building on (Hinton et al., 2015; Shenfeld et al., 2026; Hübotter et al., 2026), we use knowledge distillation to internalize the information carried by intermediate tokens into the student's parameters. We propose **masked distillation**, a knowledge-distillation framework in which a student model is trained to predict only the solution tokens conditioned on the question, while the teacher provides feedback on the student's responses after being conditioned on the question as well as its CoT trace. We instantiate this framework in two settings: (i) a *self-distillation* setting, where the same model serves as the teacher in thinking mode and as the student in non-thinking mode, and (ii) a *dual-model* setting, where a reasoning model supervises a separate non-thinking model over the solution tokens. By treating intermediate tokens as a scaffold that reasoning models use to fit over the solution tokens, we additionally study an α-*suffix masked distillation* variant where the student internalizes the $(1-\alpha)$ *prefix* of the teacher's intermediate trace and is trained to produce the remaining α-fraction *suffix* of the trace at inference, together with the solution tokens.

In this study, we explore the following research questions:

- **RQ1.** Through masked distillation, can a student model internalize the information present in the teacher's reasoning tokens and achieve similar task performance with improved inference efficiency?
- **RQ2.** If the masked-distillation student does not achieve similar task performance as the teacher, does providing additional scaffolding (measured in intermediate tokens emitted at inference) improve the task performance and what is the tradeoff against inference cost?
- **RQ3.** Does the internalization of intermediate tokens lead to degradation of performance on out-of-distribution (OOD) tasks, and does providing additional scaffolding help the student maintain OOD performance?
- **RQ4.** Does the training objective matter? Specifically, can we provide different scaffolds to the student and can we use supervised fine-tuning on teacher rollouts to achieve similar results?

We conduct extensive experiments across two domains, Math (GSM8K, with MATH-500 and AIME-25 as out-of-distribution splits) and Countdown (with target-range and search-depth shifts as out-of-distribution splits) under both *self-distillation* and *dual-model* settings. Sweeping the suffix-scaffold parameter $\alpha \in \{0, 0.3, 0.5, 0.7, 1.0\}$, our results show that the ability to fully internalize the information present in intermediate tokens is task-dependent: it succeeds on GSM8K, where the base student has prior exposure to the domain, but fails on Countdown. However, our results demonstrate that the α-suffix scaffold closes this gap at modest cost; for example, applying $\alpha = 0.3$ to self-distillation on Countdown raises accuracy from 41.7% (fully masked) to 86.2%, essentially matching the teacher (87.3%) at a $\sim 1.3\times$ reduction in inference tokens relative to the non-masked variant. In out-of-distribution tasks, the models

trained under various scaffold paradigms transfer cleanly under target-range shift, and in the self-distillation setting they generalize better than the non-masked variant under search-depth shift. These findings establish suffix scaffolding as a controlled axis along which the accuracy vs. inference-cost trade-off can be tuned, with the optimal operating point determined by the task and the teacher-student capacity gap.

The rest of the paper is organized as follows. Section 2 provides the background on knowledge-distillation techniques. Section 3 presents the masked-distillation framework. Section 4 details the experimental setup. Section 5 reports results on GSM8K and Countdown and discusses their implications for the scaffolding–generalization tradeoff.

2. Background

2.1. Knowledge Distillation

Traditionally, Knowledge distillation is a framework where there is a student-teacher pair and the student model is being trained to mimic the behaviour of the teacher by minimizing the divergence between their output distributions. This framework was first introduced by (Hinton et al., 2015) for transferring knowledge from an ensemble or from a large highly regularized model into a smaller, distilled model. In the context of autoregressive models, knowledge distillation has been extensively studied for training a smaller Language model to mimic the outputs of a larger teacher LLM. Let θ denote the student model's parameters, and p_S^θ denote the student model's policy, differentiable w.r.t θ. Let us denote the dataset of input-output pairs as (X, Y). For a divergence D, we define the discrepancy between token-level distributions of p_T and p_S as

$$D_{p_T \| p_S^\theta}(y \mid x) := \frac{1}{L_y} \sum_{n=1}^{L_y} D\big(p_T(\cdot \mid y_{<n}, x) \,\big\|\, p_S^\theta(\cdot \mid y_{<n}, x)\big)\,, \tag{1}$$

There are many variants of KD studied in the literature and the distinction between the variants is based on whether the teacher's probability distribution p_T is accessible during training and from where the targets, Y, are obtained.

- **Sequence-level KD:** (Kim & Rush, 2016) If p_T is not accessible during training, and the outputs Y are sampled from the teacher on inputs X, then with supervised finetuning, the student model is trained to maximize the likelihood of high-probability teacher sequences.
- **Supervised KD:** (Hinton et al., 2015), (Sanh et al., 2019) The student model is trained to mimic the next-token probability distribution of the teacher model. The loss $L_{\mathrm{SD}}(\theta)$ over the dataset (X, Y) is calculated as

$$\min_\theta \; L_{\mathrm{SD}}(\theta) := \mathbb{E}_{(x,y)\sim(X,Y)} \left[D_{\mathrm{KL}}\big(p_T \,\big\|\, p_S^\theta\big)\,(y \mid x) \right]. \tag{2}$$

- **On-policy KD:** MinilLM (Gu et al., 2024), GKD (Agarwal et al., 2024), performs on-policy knowledge distillation by generating responses from the student model.

$$L_{\mathrm{GKD}}(\theta) := \mathbb{E}_{x\sim X,\; y\sim p_S^\theta(\cdot|x)} \left[D_{\mathrm{KL}}\big(p_T(\cdot \mid y_{<n}, x) \,\big\|\, p_S^\theta(\cdot \mid y_{<n}, x)\big) \right] \tag{3}$$

  In (Agarwal et al., 2024), authors have shown that on-policy KD works well with different variants of KL divergence such as forward KL, reverse KL and JSD. Sequence-level KD and Supervised KD have the problem of distribution mismatch between output sequences seen during training and those generated by the student during inference. The on-policy training in GKD addresses this drawback effectively.

3. Masked Distillation

We present a knowledge-distillation framework, which we call *masked distillation*, that internalizes the intermediate-token (IT) generation process (or the so-called reasoning process) of a reasoning model π^T (the teacher model) into a non-thinking student model π^S. The goal is to push the teacher's reasoning into the student's parameters so that, at inference time, the student directly generates the solution tokens conditioned only on the input problem, without emitting any intermediate tokens of its own. This makes inference substantially more efficient and cheaper, since the

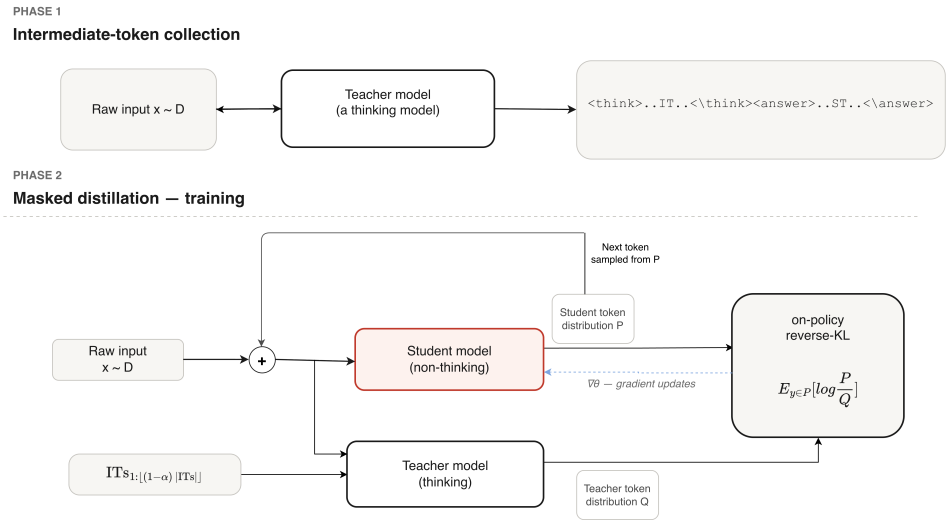


Figure 1. **Overview of the masked distillation framework.** *Phase 1, intermediate-token collection (top):* for each input question $x \sim \mathcal{D}$ we sample a response from the teacher π^T. The response has the form *<think>(ITs) <\think><answer>(STs)<\answer>*, from which we extract the intermediate tokens (ITs) and the solution tokens (STs). *Phase 2, training (bottom):* the non-thinking student π_θ^S is conditioned only on x, while the frozen teacher is conditioned on x together with the first $(1-\alpha)$ fraction of the ITs from Phase 1. The parameter α controls how much of the intermediate trace the student is supervised to emit at inference: the student is trained to reproduce the last α fraction of the teacher's intermediate trace together with the solution tokens. The student's next-token distribution P is matched against the teacher's distribution Q via the on-policy reverse-KL loss $\mathbb{E}_{y\sim P}[\log P/Q]$ computed on student-sampled tokens; gradients update only the student. At $\alpha = 0$ the variant reduces to fully masked distillation, and at $\alpha = 1$ to non-masked distillation. When the teacher and the student are the same model, this is the *self-distillation* setting; when the teacher is a larger thinking model and the student is a smaller non-thinking model, this is the *dual-model* setting.

intermediate computation has already been internalized into the student's internal representations during distillation. Concretely, the student is trained to mimic the teacher's conditional distribution over the input problem x and the teacher's own generated intermediate tokens, $\pi^T(\cdot \mid x, \mathrm{ITs})$.

We consider two settings of *masked distillation*: *self-distillation* and the *dual-model* setting. In the *self-distillation* setting, the same model acts as both the teacher and the student: the teacher operates in thinking mode and generates intermediate tokens, while the student operates in non-thinking mode and is trained to directly produce the final solution. In the *dual-model* setting, a larger reasoning model serves as the teacher and supervises a separate, smaller non-thinking student on the generation of solution tokens. Unless otherwise specified, we use the term *masked distillation* to refer generically to the framework, and explicitly distinguish between the two settings only when necessary.

Figure 1 shows an overview of the masked distillation framework. In **Phase 1**, we collect intermediate tokens for each question $x \in \mathcal{D}$ by sampling a response from the teacher model and extracting the segment between the `<think>` and `</think>` tags as the ITs; In **Phase 2**, we train the student model by minimizing the divergence between the student distribution $\pi_\theta^S(\cdot \mid x)$ and the teacher distribution $\pi^T(\cdot \mid x, \mathrm{ITs})$, where the teacher is conditioned on x together with the ITs it generated in Phase 1, while the student is conditioned only on x. The masked-distillation objective is the reverse-KL divergence between the student and teacher next-token distributions, computed on student-sampled responses. The loss is defined as

$$\begin{aligned}\mathcal{L}_{\mathrm{MD}} &= D_{\mathrm{KL}}\big(\pi_\theta^S(\cdot \mid x, y_{<t}) \,\big\|\, \pi^T(\cdot \mid x, \mathrm{ITs}, y_{<t})\big) \\ &= \mathbb{E}_{y_t' \sim \pi_\theta^S(\cdot \mid x, y_{<t})}\left[\log \frac{\pi_\theta^S(y_t' \mid x, y_{<t})}{\pi^T(y_t' \mid x, \mathrm{ITs}, y_{<t})}\right] \\ &= \sum_{y_t' \in \mathcal{V}} \pi_\theta^S(y_t' \mid x, y_{<t}) \, \log \frac{\pi_\theta^S(y_t' \mid x, y_{<t})}{\pi^T(y_t' \mid x, \mathrm{ITs}, y_{<t})}.\end{aligned} \tag{4}$$

where $\mathcal{V}$ is the vocabulary. Since we use the same family of models for the teacher and the student, their vocabularies are identical.

We compare masked distillation against a *non-masked* variant in which both the teacher and the student are conditioned only on the input question $x \in \mathcal{D}$, so that the student also learns to reproduce the teacher's intermediate-token generation process before emitting the solution. For non-masked distillation the loss is

$$\mathcal{L}_{\text{Non-MD}} = D_{\text{KL}}\big(\pi_\theta^S(\,\cdot \mid x, y_{<t}) \,\big\|\, \pi^T(\,\cdot \mid x, y_{<t})\big)\,. \tag{5}$$

In the *self-distillation* setting, since the teacher and the student are the same model, the non-masked variant reduces to running the teacher in thinking mode at inference.

Masked and non-masked distillation lie at the two extremes of a spectrum that controls how much of the teacher's reasoning process the student reproduces at inference. Under masked distillation the student fully internalizes the intermediate-token generation process and emits only the solution tokens at inference; under non-masked distillation the student, like the teacher, first produces intermediate tokens and only then the final answer. To probe the effect of intermediate-token scaffolding between these extremes, we additionally study an α-*suffix masked distillation* variant where the student internalizes the $(1-\alpha)$ *prefix* of the teacher's intermediate trace and is trained to produce the remaining α-fraction *suffix* of the trace at inference, together with the solution tokens. The teacher is conditioned on input question and prefix that the student has to internalized, while the student is conditioned only on input question:

$$\mathcal{L}^{\alpha}_{\text{MD-}Sfx} = D_{\text{KL}}\big(\pi_\theta^S(\,\cdot \mid x, y_{<t}) \,\big\|\, \pi^T\big(\,\cdot \,\big|\, x,\ \text{ITs}_{1:\lfloor(1-\alpha)\,|\text{ITs}|\rfloor},\ y_{<t}\big)\big)\,, \tag{6}$$

where $\text{ITs}_{1:\lfloor(1-\alpha)\,|\text{ITs}|\rfloor}$ denotes the first $(1-\alpha)$ tokens of the teacher's intermediate trace. At $\alpha = 0$ the variant reduces to fully masked distillation (Eq. 4); at $\alpha = 1$ it reduces to non-masked distillation (Eq. 5). In practice we report $\alpha \in \{0.3, 0.5, 0.7\}$ to map the suffix-budget axis at a few representative operating points.

4. Experimental Setup

We conducted our experiments in the following domains -

1. **Math -** We used *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 generalization in this domain, we test on the AIME-25 (on Hugging Face, 2025) and MATH-500 (Lightman et al., 2024) datasets as OOD splits that share the same arithmetic substrate but exercise it at substantially higher difficulty.

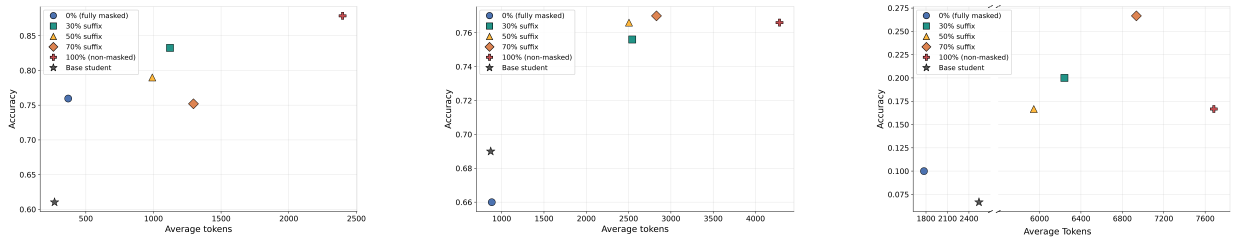
2. **Countdown -** This domain (Countdown.) is a generalized version of the classic 24 Game (Yang et al., 2022), where the objective is to combine a set of input numbers using basic arithmetic operations (+, –, ×, ÷) to reach a specified target number. In our ID dataset, each problem consists of 3 to 4 two-digit input numbers, with the target number also being a two-digit number. The dataset contains 9K examples, split into 8K training instances and 1K test instances. For OOD testing, we have three datasets, each perturbing a different axis. *OOD-1* keeps the input cardinality and digit count fixed but shifts the target to a one-digit number (*target-range shift*; 678 problems). *OOD-2* and *OOD-3* keep the digit counts fixed but increase the input cardinality to 5 and 6 numbers respectively (*search-depth shift*; 522 and 478 problems).

Training details: We run all experiments on two NVIDIA H100 GPUs with 80 GB of VRAM each, and our implementation builds on the `verl` library (Sheng et al., 2024). For our experiments in the Math domain, in the *self-distillation* setting, we use the Qwen3-1.7B (Yang et al., 2025) base model where the model in non-thinking mode serves as the student, while the same model in thinking mode acts as the teacher. For our *dual-model* setting, we use the Qwen3-1.7B base model in thinking mode as the teacher and Qwen2-0.5B-Instruct (Yang et al., 2024) as the student model. Similarly in the countdown domain, we use Qwen3-4B base model as the teacher and Qwen2-0.5B-Instruct model as the student. The masked-distillation loss is the reverse-KL divergence between the student and teacher next-token distributions, computed over the full vocabulary. We provide additional hyperparameter details in the appendix section A.

5. Results and Discussion

Figures 2–5 show the accuracy and average response length for the different values of α in our suffix-scaffold masked-distillation variants, evaluated on GSM8K and Countdown across in-distribution (ID) and out-of-distribution (OOD)

splits. Referring to these, we address our first three research questions: RQ1 (the effect of fully internalizing the teacher's intermediate tokens on in-distribution performance) in subsection 5.1; RQ2 (whether additional scaffolding budget improves task performance, and its trade-off against inference cost) in subsection 5.2; and RQ3 (Does internalizing intermediate tokens degrade OOD performance, and whether scaffolding helps to recover it) in subsection 5.3. Finally, we compare on-policy reverse-KL distillation with supervised fine-tuning on teacher rollouts (RQ4) in subsection 5.4.



(a) GSM8K (in-distribution) *(b)* MATH-500 (OOD) *(c)* AIME25 (OOD)

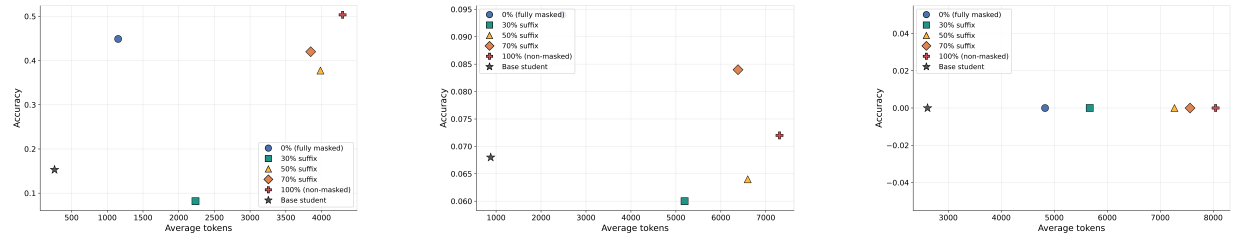
Figure 2. Accuracy vs. average response length on math domain (self-distillation setting) for suffix-scaffold masked distillation variants with $\alpha \in \{0, 0.3, 0.5, 0.7, 1.0\}$ Teacher: Qwen3-1.7B (thinking); Student: Qwen3-1.7B (non-thinking).

5.1. Effects of full internalization of teacher's intermediate tokens on in-distribution performance

In this section we analyze whether student can internalize the information present in the teacher's reasoning tokens while still matching teacher-level performance at lower inference cost. The degree to which full internalisation succeeds depends on (i) distillation setting (self-distillation vs. dual-model) and (ii) task. We discuss the two settings in turn.

Self-distillation setting: On GSM8K (Figure 2a), the base student (Qwen3-1.7B-Base in non-thinking mode) attains 61.0% accuracy on the GSM8K test set, while the teacher (Qwen3-1.7B-Base in thinking mode) reaches 87.9%. Training using the fully masked distillation ($\alpha = 0$) paradigm closes this gap: the student reaches 76.0% accuracy while emitting only 369.6 tokens on average against the teacher's 2398.1, a $6.5\times$ reduction in inference cost.

On Countdown (Figure 4a), the base student (Qwen3-4B-Base in non-thinking mode) attains 56.2% accuracy on the Countdown ID test set and the teacher (Qwen3-4B-Base in thinking mode) reaches 87.3%. Unlike on GSM8K, fully masked distillation here does not recover this gap with the fully masked student reaching only 41.7% accuracy. The student does emit far shorter responses on average (96.3 tokens against the teacher's 2499.6, a $26\times$ reduction), but its in-distribution accuracy is now lower than the student's base model.



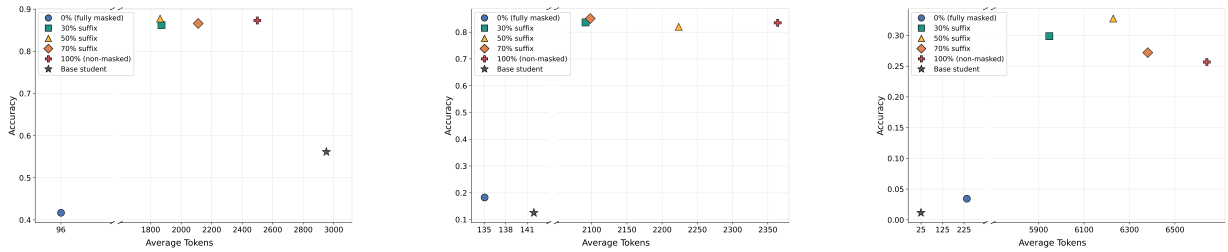
(a) GSM8K (in-distribution) *(b)* MATH-500 (OOD) *(c)* AIME25 (OOD)

Figure 3. Accuracy vs. average response length on the math domain (dual-model setting), for suffix-scaffold masked distialltion variants with $\alpha \in \{0, 0.3, 0.5, 0.7, 1.0\}$. Teacher: Qwen3-1.7B (thinking); Student: Qwen2-0.5B-Instruct.

Dual-model setting: On GSM8K (Figure 3a), the student (Qwen2-0.5B-Instruct) is distilled from a larger Qwen3-1.7B teacher that reaches 89.23% accuracy. The base student already attains 15.69% on GSM8K without any distillation, indicating that GSM8K style problems are likely well-represented in its pretraining distribution. Full internalization substantially improves the student over its base model, raising accuracy from 15.69% to 45.26% at a $5\times$ reduction in average response length relative to the teacher. The performance of the fully-masked student is comparable to the performance of the model trained with non-masked distillation with a significant gain in inference efficiency.

On Countdown (Figure 5a), the student is distilled from a larger Qwen3-4B teacher that reaches 87.3% accuracy.

Notably, the base student has 0% accuracy on Countdown which suggests that the Countdown puzzle format is likely not well represented in its pretraining data. Fully masked distillation increases the accuracy of the student from 0% to 34.1%, but unlike the GSM8K dual-model setting, there is a significant gap in performance of the student and of the non-masked variant (81.1%) which is trained to reproduce the full thinking trace. The fully masked student does emit much shorter responses on average (403.8 tokens against the non-masked variant's 3052.6, a $7.6\times$ reduction).



(a) In-distribution *(b)* OOD: target shift *(c)* OOD: 5 numbers

Figure 4. Accuracy vs. average response length on Countdown (self-distillation setting), for suffix-scaffold masked distillation variants with $\alpha \in \{0, 0.3, 0.5, 0.7, 1.0\}$. Teacher: Qwen3-4B (thinking); Student: Qwen3-4B (non-thinking).

5.2. Effect of providing additional scaffolding, α-suffix, on task performance and inference cost

The RQ1 results show that full internalization does not always reach the same performance as the model in thinking mode or the non-masked variant. A natural follow-up is whether allowing the student to emit a partial scaffold at inference, a fraction α of the teacher's intermediate trace can recover this accuracy, and at what cost in inference budget. In addition to the fully masked and non-masked variants, we sweep $\alpha \in \{0.3, 0.5, 0.7\}$ on both tasks and trace the resulting accuracy vs. response-length frontier.

Self-distillation setting: On GSM8K (Figure 2a), adding even a small suffix scaffold delivers most of the recovery. With $\alpha = 0.3$ the student reaches 83.2% accuracy, a 7-point gain over the fully masked 76.0% baseline, while emitting $1{,}122$ tokens on average. Further increasing the suffix budget leaves accuracy in a similar range (79.0% at $\alpha = 0.5$ and 75.2% at $\alpha = 0.7$), and only the full non-masked variant ($\alpha = 1.0$) reaches 87.9% at $2{,}398$ tokens. The most efficient operating point is $\alpha = 0.3$: it produces $\sim 2\times$ shorter responses than the non-masked variant ($1{,}122$ vs. $2{,}398$ tokens) at the cost of about 5 accuracy points (83.2% vs. 87.9%).

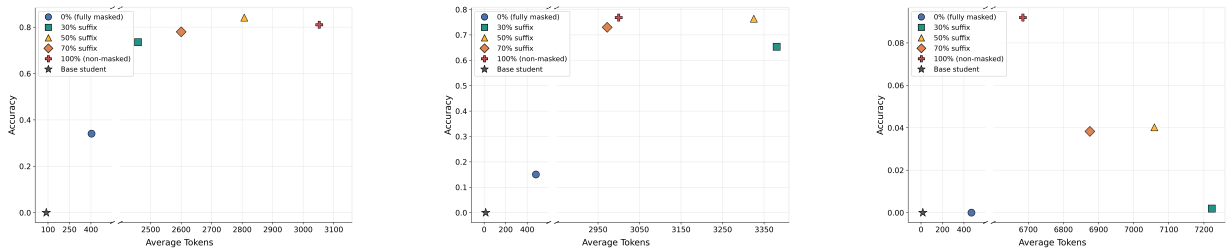
On Countdown (Figure 4a), scaffolding has a much sharper effect. The fully masked baseline reaches only 41.7%, below the base student but $\alpha = 0.3$ jumps to 86.2%, essentially matching the teacher's 87.3%. The sweep then plateaus: 87.8% at $\alpha = 0.5$, 86.6% at $\alpha = 0.7$, and 87.3% at $\alpha = 1.0$, all within a 2-point range. The most efficient operating point is therefore $\alpha = 0.3$, which recovers teacher-level accuracy at a $\sim 1.3\times$ reduction in inference cost relative to the non-masked variant ($1{,}871$ vs.$2{,}500$ tokens).

Dual-model setting: On GSM8K (Figure 3a), the fully masked variant reaches 44.9% accuracy and the non-masked variant reaches 50.3%. The narrow 5.4-point gap between these two endpoints indicates that the dual-model GSM8K setting has limited room for intermediate scaffolding to help: the residual that an α-suffix variant could close is small, and the inference-cost penalty of any non-zero scaffold rapidly outweighs the available accuracy gain.

On Countdown (Figure 5a), scaffolding produces a progressive and substantial recovery. From the fully masked baseline of 34.1%, accuracy rises to 73.5% at $\alpha = 0.3$, 84.2% at $\alpha = 0.5$, 78.0% at $\alpha = 0.7$, and 81.1% at $\alpha = 1.0$. The peak of the sweep is $\alpha = 0.5$, which slightly exceeds the non-masked baseline (84.2% vs. 81.1%), and the response length at this operating point ($2{,}807$ tokens) is below the non-masked variant's $3{,}053$. The dual-model Countdown frontier is therefore not just recovered but improved by the suffix scaffold: $\alpha = 0.5$ is pareto-optimal than the non-masked variant on both accuracy and inference cost.

5.3. Generalization under distribution shift

The RQ1 and RQ2 results analyze the accuracy v/s efficiency trade-off on in distribution tasks. In this section, we analyze whether the various training paradigms demonstrate similar trends in OOD settings (RQ3).



(a) In-distribution *(b)* OOD: target shift *(c)* OOD: 5 numbers

Figure 5. Accuracy vs. average response length on Countdown (dual model setting), for suffix-scaffold masked distillation variants with $\alpha \in \{0, 0.3, 0.5, 0.7, 1.0\}$. Teacher: Qwen3-4B (thinking); Student: Qwen2-0.5B-Instruct.

Self-distillation setting: On the math domain (Figure 2b), the suffix sweep on MATH-500 broadly follows the GSM8K ID pattern. The base student attains 69.0% on MATH-500 and the fully masked variant ($\alpha = 0$) reaches only 66.0% slightly below base but every scaffolded variant lifts accuracy above the base model: 75.6% at $\alpha = 0.3$, 76.6% at $\alpha = 0.5$, 77.0% at $\alpha = 0.7$, and 76.6% at $\alpha = 1.0$ (non-masked). The $\alpha = 0.3$ operating point recovers essentially all of the non-masked accuracy at a $\sim 1.7\times$ shorter response length (2,542 vs. 4,284 tokens). On AIME-25 (Figure 2c) the base student reaches 6.7%, the fully masked variant 10.0%, and the scaffold variants range from 16.7% to 26.7% in a non-monotonic pattern. Interestingly, we see that the $\alpha = 0.7$ variant performs better than the teacher model while being more efficient during inference.

On Countdown (Figures 4b and 4c), the suffix scaffold transfers cleanly under target-range shift but produces a more interesting pattern under search-depth shift. Under target-range shift, the base student reaches 12.5% and the fully masked variant 18.3%; the scaffold variants close most of the gap to non-masked, with 83.6% at $\alpha = 0.3$, 82.0% at $\alpha = 0.5$, 85.1% at $\alpha = 0.7$, and 83.5% at $\alpha = 1.0$. The $\alpha = 0.3$ variant essentially matches the non-masked baseline at this shift. Under search-depth shift, accuracy is much lower across the board (1.1% base, 3.4% fully masked), but the scaffold variants actually *outperform* the non-masked baseline: 29.9% at $\alpha = 0.3$, 32.8% at $\alpha = 0.5$, 27.2% at $\alpha = 0.7$, and 25.7% at $\alpha = 1.0$. The peak $\alpha = 0.5$ variant is about 7 points above non-masked, suggesting that under search-shift OOD, the scaffold-trained student generalizes better than the teacher model.

Dual-model setting: In math domain (Figure 3b), none of the distillation variants meaningfully outperform the base student on MATH-500. The base student attains 6.8% accuracy and all distilled variants sit in a narrow 6.0%–9.4% band (9.4% fully masked, 6.0% at $\alpha = 0.3$, 6.4% at $\alpha = 0.5$, 8.4% at $\alpha = 0.7$, 7.2% at $\alpha = 1.0$). The fully masked variant is the highest of the distilled variants but only marginally above base. On AIME-25 (Figure 3c), every variant including the base student reaches 0%, These results indicate that on harder math benchmarks, the capacity gap between the Qwen3-1.7B teacher and the Qwen2-0.5B-Instruct student is large enough that neither full internalization nor suffix scaffolding recovers OOD performance: the student converges to roughly the base-student floor in every case.

On Countdown (Figures 5b and 5c), the suffix scaffold transfers under target-range shift but fails under search-depth shift. Under target-range shift, the base student reaches 0.0%, the fully masked variant 15.0%, and the scaffolded variants 65.3% ($\alpha = 0.3$), 76.4% ($\alpha = 0.5$), 73.0% ($\alpha = 0.7$), and 76.8% ($\alpha = 1.0$). The $\alpha = 0.5$ variant essentially matches the non-masked baseline, mirroring the ID pareto-optimal behaviour we observed in RQ2. Under search-depth shift, however, accuracy collapses across the board: 0.0% base, 0.0% fully masked, 0.2% at $\alpha = 0.3$, 4.0% at $\alpha = 0.5$, 3.8% at $\alpha = 0.7$, and 9.2% at $\alpha = 1.0$. Unlike the self-distillation Countdown case, the non-masked variant is the best of the distilled variants under search-depth shift, but even its accuracy only reaches to 9.2% .

5.4. Ablation study: Providing different types of scaffolds and training objectives

We investigate whether different scaffolds help the student to fit over the teacher's distribution and whether, instead of knowledge distillation with reverse KL, supervised fine-tuning (SFT) can achieve similar results.

Masked Distillation using teaching tokens: Instead of just producing the solution tokens, we prompt the teacher model to produce *teaching-tokens* which can be used to train a student model along with the solution tokens. The length of these teaching tokens can be controlled and thus, we use that as lever to control the amount of scaffolding provided to the student. We provide detailed description of teaching tokens and how they are generated in Appendix B.1. We perform

this ablation in the *dual-model* setting in the countdown domain. Our results, refer Table 1a, demonstrate that the scaffolding provided in the form of teacher tokens improves task performance as compared to full masked-distillation. However, the α-*suffix masked distillation* variants had higher task performance compared to the models trained on teaching tokens. Thus, while scaffolds generally help, not all are equally effective.

Finetuning on teacher traces: Our results show that teacher-token scaffolding does not help under SFT (refer Table 1b. The fully masked SFT student reaches 38.4% on the in-distribution split; the 100 and 1000-TT variants reach 38.5% and 30.8% respectively. This is in sharp contrast to the reverse-KL setting, where the same scaffolding sweep nearly doubles in-distribution accuracy (from 34.08% to 67.25% / 65.04%). SFT does not learn to exploit the 100 or 1000-TT scaffold, while reverse-KL does.

Table 1. **Test accuracy on Countdown.** One in-distribution (ID) split—3–4 two-digit input numbers with a two-digit target—and three out-of-distribution (OOD) splits that each perturb a single axis: target-range shift (3–4 nums, one-digit target) and search-depth shift (5 and 6 input numbers). Values in parentheses under each header give the size N of that split.

Methods	In-Distribution	Out-of-Distribution		
	(3–4 nums, 2d tgt) (1024)	(3–4 nums, 1d tgt) (678)	(5 nums) (522)	(6 nums) (478)
MD	34.08%	15.04%	0.00%	0.00%
MD (100 TTs)	67.25%	44.25%	0.57%	0.00%
MD (1k TTs)	65.04%	50.74%	0.96%	0.00%
Non-MD	81.05%	76.84%	9.20%	0.00%

(a) Reverse-KL distillation.

Methods	In-Distribution	Out-of-Distribution		
	(3–4 nums, 2d tgt) (1024)	(3–4 nums, 1d tgt) (678)	(5 nums) (522)	(6 nums) (478)
Masked-SFT	38.4%	10.0%	0.0%	0.0%
Masked-SFT (100 TTs)	38.5%	9.9%	0.0%	0.0%
Masked-SFT (1k TTs)	30.8%	12.2%	0.0%	0.0%
Non-masked SFT	75.7%	69.3%	2.3%	0.0%

(b) Supervised fine-tuning (SFT).

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 distillation**, a knowledge-distillation framework in which a student model is trained to predict only the solution tokens conditioned on the question, while the teacher provides feedback on the student's responses after being conditioned on the question as well as its CoT trace. By treating intermediate tokens as a scaffold that reasoning models use to fit over the solution tokens, we also studied α-*suffix masked distillation* variants where student internalizes the $(1-\alpha)$ *prefix* of teacher's intermediate trace and is trained to produce the remaining α-fraction *suffix* of the trace at inference, together with solution tokens. We conduct extensive experiments across two domains, Math and Countdown under both *self-distillation* and *dual-model* settings. Sweeping the suffix-scaffold parameter $\alpha \in \{0, 0.3, 0.5, 0.7, 1.0\}$, our results show that the ability to fully internalize the information present in intermediate tokens is task-dependent: it succeeds on GSM8K, where base student has prior exposure to the domain, but fails on Countdown. However, our results demonstrate that the α-suffix scaffold closes this gap at modest cost; for example, applying $\alpha = 0.3$ to self-distillation on Countdown raises accuracy from 41.7% (fully masked) to 86.2%, essentially matching the teacher (87.3%) at a $\sim 1.3\times$ reduction in inference tokens relative to the non-masked variant. In out of distribution tasks, models trained under various scaffold paradigms transfer cleanly under target-range shift, and in the self-distillation setting they generalize better than the non-masked variant under search-depth shift. These findings establish suffix scaffolding as a controlled axis along which the accuracy vs. inference-cost trade-off can be tuned, with the optimal operating point determined by the task and the teacher-student capacity gap.

Future Work All variants in our experiments use the same amount of training data and a similar number of optimization steps, which is a conservative setting for the masked variants given their sparser per-question supervision. Matching the effective supervision across variants is left for future work. In addition, we want to examine whether the amortization tradeoff between the additional training cost via Masked Distillation and the inference-time savings is favorable. Finally, the masking objective is agnostic to the source of the intermediate tokens. The same recipe can be applied when the intermediate signal is provided through environment feedback rather than a teacher model, as in SDPO (Hübotter et al., 2026). We plan to investigate whether the source of the intermediate signal (teacher versus environment) affects the scaffolding–generalization tradeoff observed in our experiments.

Acknowledgments

This research is supported in part by grants from ONR (N00014-25-1-2301 and N00014-23-1-2409), DARPA (HR00112520016), DoD RAI (via CMU subcontract 25-00306-SUB-000), and a generous gift from Qualcomm. We also thank Cloudexe for the GPU Catalyst Fellowship.

References

Agarwal, R., Vieillard, N., Zhou, Y., Stanczyk, P., Ramos Garea, S., Geist, M., and Bachem, O. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

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

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

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

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

Deng, Y., Prasad, K., Fernandez, R., Smolensky, P., Chaudhary, V., and Shieber, S. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023.

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

Gu, Y., Dong, L., Wei, F., and Huang, M. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Hübotter, J., Lübeck, F., Behric, L., Baumann, A., Bagatella, M., Marta, D., Hakimi, I., Shenfeld, I., Buening, T. K., Guestrin, C., et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Kambhampati, S., Stechly, K., Valmeekam, K., Saldyt, L. P., Bhambri, S., Palod, V., Gundawar, A., Samineni, S. R., Kalwar, D., and Biswas, U. Stop anthropomorphizing intermediate tokens as reasoning/thinking traces! In *NeurIPS 2025 Workshop on Bridging Language, Agent, and World Models for Reasoning and Planning*, 2025.

Kim, Y. and Rush, A. M. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Luo, H., Shen, L., He, H., Wang, Y., Liu, S., Li, W., Tan, N., Cao, X., and Tao, D. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025.

Ma, X., Wan, G., Yu, R., Fang, G., and Wang, X. 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.

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

Palod, V., Valmeekam, K., Stechly, K., and Kambhampati, S. Performative thinking? the brittle correlation between cot length and problem complexity. *arXiv preprint arXiv:2509.07339*, 2025.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Shenfeld, I., Damani, M., Hübotter, J., and Agrawal, P. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Shrivastava, V., Awadallah, A., Balachandran, V., Garg, S., Behl, H., and Papailiopoulos, D. Sample more to think less: Group filtered policy optimization for concise reasoning. *arXiv preprint arXiv:2508.09726*, 2025.

Snell, C., Klein, D., and Zhong, R. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Team, K., Du, A., Gao, B., Xing, B., Jiang, C., Chen, C., Li, C., Xiao, C., Du, C., Liao, C., et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Valmeekam, K., Stechly, K., Palod, V., Gundawar, A., and Kambhampati, S. Beyond semantics: The unreasonable effectiveness of reasonless intermediate tokens. *arXiv preprint arXiv:2505.13775*, 2025.

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Yang, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Liu, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Guo, Z., and Fan, Z. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yang, M. S., Schuurmans, D., Abbeel, P., and Nachum, O. Chain of thought imitation with procedure cloning. *Advances in Neural Information Processing Systems*, 35:36366–36381, 2022.

A. Hyperparameter details

In our masked-distillation experiments, the student uses a maximum prompt length of $2{,}048$ tokens and a maximum response length of $8{,}192$ tokens. The teacher uses a longer prompt length of $10{,}000$ tokens, since its prompt is augmented with the intermediate tokens (ITs) collected in Phase 1. Each training step uses a question batch size of 16, a mini-batch size of 1, and 8 rollouts per question. Rollouts at training time are sampled with the vLLM inference engine at temperature 1.0. Validation uses one rollout per question with temperature 0.6 and top-$p = 0.95$. We optimize with AdamW at a constant learning rate of 1×10^{-5}.

B. What are teaching tokens?

In *fully masked* distillation, Phase 1 prompts the teacher with the raw input question only, eliciting a response of the form `<think>ITs</think><answer>STs</answer>`, from which we extract the intermediate tokens. With the fully-masked objective (Eq. 4), the student therefore learns to generate solution tokens only. For the teaching-token variants, in Phase 1 we additionally pass an explicit instruction prompt to the teacher, asking it to summarize its own intermediate tokens into a concise scaffold of $k \in \{100, 1000\}$ tokens (the *teaching tokens*, TTs) that the student can use as a hint while producing the solution. The teacher's response then takes the form `<think>ITs</think><teaching>TTs</teaching><answer>STs</answer>` (the instruction prompt and statistics on the generated teaching tokens are reported in Appendix .) During Phase 2 the teacher remains conditioned on the question, the instruction prompt, and the ITs, while the student now learns to emit the teaching tokens followed by the solution tokens.

B.1. Prompt templates and teaching-token statistics

In Phase 1 (Section 3), we collect teacher intermediate tokens by prompting the teacher with one of two templates, depending on the distillation variant. The student receives only the question; the teacher receives the question alone for fully masked distillation ($k = 0$), and the question augmented with an *instruction prompt* that elicits teaching tokens for the $k \in \{100, 1{,}000\}$ variants. Below we show a representative Countdown example.

Prompt template (Countdown)

Using the numbers [10, 8, 27, 9], create an equation that equals 14. You can use basic arithmetic operations (+, -, *, /) one or multiple times but each number can only be used once. Return the final equation in `<answer> </answer>` tags, for example `<answer> (1 + 2) / 3 </answer>`.

You are a teacher. First, solve the problem. Then, write a teaching explanation (approximately $\{100,\ 1{,}000\}$ words) that captures the key insights and reasoning steps a student would need to solve this problem on their own. Do not show your full working—distill it into useful teaching points. Finally, give your final answer in `<answer> </answer>` tags.

Format your response as:

```
<teaching>
[Your teaching explanation here]
</teaching>
<answer> [result] </answer>
```

The unhighlighted text is the question; it is given to the student at training and inference, and to the teacher under fully masked distillation ($k = 0$). The highlighted block is the instruction prompt; it is appended to the question only for the teaching-token variants ($k = 100$ and $k = 1{,}000$), with the target word count set to 100 or $1{,}000$ respectively. Under this prompt, the teacher emits a response of the form `<think>...ITs...</think><teaching>...TTs... </teaching><answer>...STs...</answer>`; under the question-only prompt, it emits `<think>...ITs...</think><answer>...STs...</answer>`.

Generated teaching-token length statistics

Table 2 reports word-count statistics for the teaching tokens (TTs) generated by the teacher under each instruction-prompt variant on Countdown. Both variants undershoot the target word count—the teacher tends to write tighter teaching blocks than asked—but the two distributions remain well separated, and the modal ranges align with the intent of each variant.

Table 2. **Teaching-token length, in words, generated by the teacher under each instruction-prompt variant** (Countdown). The "target" column gives the word count requested in the instruction prompt.

Variant	Target	Mean	Median	Modal range
$k = 100$ TTs	$\sim$100 words	80	73	50–99 words
$k = 1{,}000$ TTs	$\sim$1,000 words	389	383	300–499 words

C. SFT variant

We additionally study a supervised fine-tuning (SFT) variant of masked distillation, in which the student is trained on teacher responses with the standard cross-entropy loss:

$$\mathcal{L}_{\mathrm{SFT}} = -\,\mathbb{E}_{x\sim\mathcal{D},\; y\sim\pi^{T}(\,\cdot\,|\,x,\,\mathrm{ITs})}\left[\sum_{t=1}^{|y|} m_t\,\log \pi_\theta^{S}(y_t \mid x, y_{<t})\right], \tag{7}$$

where $m_t \in \{0, 1\}$ is a per-token mask that selects which positions in the teacher response contribute to the loss. The mask realises the same scaffolding spectrum as the reverse-KL objective: $m_t = 1$ on solution tokens only for the fully masked variant; $m_t = 1$ on the (teaching $\oplus$ solution) span for the masked-k variants; and $m_t = 1$ on the full (intermediate $\oplus$ solution) span for non-masked SFT. We use exactly the teacher rollouts collected in Phase 1, so the only differences between the SFT and reverse-KL variants are the loss form (forward vs. reverse KL on a single sample) and the source of the gradient (teacher rollouts vs. student-sampled responses).

Hyperparameter details: For the SFT variant on Countdown, we fine-tune the student, Qwen2-0.5B-Instruct, on solution traces produced by the teacher, Qwen3-4B operating in reasoning mode. Teacher traces were generated by sampling at temperature 0.6 with a maximum generation length of 8192 tokens, and the student was trained in non-thinking mode. We use parameter-efficient fine-tuning via LoRA (Hu et al., 2022), with rank r=64 and scaling factor $\alpha = 128$. For each variant, we determine the learning rate, and batch size after a hyperparameter sweep.

Per-variant configurations

- **Non-masked SFT (thinking-mode student):** learning rate 1×10^{-4}, batch size 4,
- **Masked-SFT (1k TTs):** learning rate 1×10^{-4}, batch size 4,
- **Masked-SFT (100 TTs):** learning rate 5×10^{-4}, batch size 4,
- **Masked-SFT :** learning rate 1×10^{-4}, batch size 8.