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

Anonymous Authors[1]

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 their length is not a reliable indicator of the true computational complexity of the problem instance. This raises a natural question: can the computation expressed in these intermediate tokens be internalized into the parameters of a smaller student model through knowledge distillation, enabling the student to produce answers directly (or with much shorter intermediate traces), and how does such internalization affect performance on out-of-distribution problems? We investigate this question through controlled distillation experiments, transferring knowledge from a Qwen3-4B thinking teacher mdoel to a Qwen2.5-0.5B-Instruct student model across two reasoning domains: GSM8K (grade-school arithmetic) and Countdown (a number-puzzle search task). We vary two key design dimensions. The first is the amount of intermediate scaffolding provided to the student during training: a non-masked regime, where the student is trained on the teacher's full thinking-plus-solution trace, and a masked regime, where the student is trained to predict the solution from the prompt along with a fixed budget of teaching tokens $k \in \{0, 100, 1000\}$ sampled from the teacher's trace. The second dimension is the training objective: reverse-KL on-policy distillation, where the student generates responses and is trained to match the teacher's response distribution, versus supervised fine-tuning (SFT) on teacher-generated rollouts (off-policy).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the FoGen Workshop at ICML 2026. Do not distribute.

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. This is consistent with theoretical results: (Merrill & Sabharwal, 2023) show that chain-of-thought strictly extends the expressive power of a fixed-depth transformer, and scaling trends on reasoning benchmarks consistently favor longer traces. The same property that makes these traces useful, however, makes them expensive at inference. 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.

Two broad lines of work try to reduce this cost. The first adds 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)). 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. The second line removes the trace altogether by internalizing the intermediate tokens into the 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 with an $11\times$ token reduction. (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 related line of work replaces explicit tokens with continuous representations. Coconut (Hao et al., 2024) feeds the model's hidden state back as the next-step input rather than decoding it; CODI (Shen et al., 2025) self-distills explicit CoT into continuous tokens; Compressed Chain-of-Thought (Cheng & Van Durme, 2024) and Token Assorted (Su et al., 2025) replace chunks of reasoning with learned or quantized representations. A separate question is whether the intermediate tokens need to carry any content or semantics at all:

beyond semantics (Valmeekam et al., 2025) (Kambhampati et al., 2025), pause tokens (London & Kanade, 2025), filler tokens (Pfau et al., 2024), and recurrent-depth transformers (Geiping et al., 2025) suggest that part of the gain may come from the extra forward-pass compute itself, independent of what those tokens mean.

Across these methods, a common pattern emerges: in-distribution accuracy can usually be matched with little or no explicit reasoning at inference, provided the training setup is right. Whether the over-training is worth the cost saving and more importantly whether the resulting students generalize, however, has received much less attention. Each method invests substantial training-time effort to push reasoning into parameters, but whether that effort transfers to out of distribution problems or not is unclear. (Zhao et al., 2025) show that in-distribution (ID) strong reasoners can collapse on out-of-distribution (OOD) inputs, so ID-only evaluation can mistake a memorized-trace student for an internalized one. (Wen et al., 2025) show that even small amounts of structured discrete scaffolding can recover OOD performance. Yet most existing internalization papers report only the endpoints (full CoTvs. no CoT) or a smooth removal schedule, and rarely measure OOD explicitly.

We address this gap by treating the amount of intermediate scaffolding as a controlled variable and measuring its effect on both in-distribution and out-of-distribution accuracy. We propose **masked distillation**: a knowledge-distillation framework where a smaller non-thinking student is trained to predict only solution tokens conditioned on the question prompt, with the teacher's intermediate tokens masked out of the student's loss. At inference, the student sees only the question and emits the answer directly. The inference saving comes from the student having absorbed the teacher's intermediate computation into its parameters during training.

We compare masked distillation against a non-masked baseline, in which the student is supervised on the full thinking trace and reproduces it at inference. Between these two endpoints we vary the scaffolding budget: the student is trained to emit $k \in \{100, 1000\}$ teaching tokens before the answer. We also vary the training objective, comparing on-policy reverse-KL distillation against supervised fine-tuning on teacher rollouts. Our experiments use a Qwen3 thinking teacher (1.7B for GSM8K, 4B for Countdown) and a Qwen2.5-0.5B-Instruct student on two reasoning domains: GSM8K and Countdown.

Within this framework we ask four research questions:

- **RQ1.** If the student internalizes the teacher's full intermediate-token generation process to improve inference efficiency, how much generalization does it lose on out-of-distribution problems relative to a non-masked student that reproduces the teacher's intermediate tokens before the answer at inference?
- **RQ2.** If masked-distillation students degrade on out-of-distribution problems relative to the non-masked variant, how much teacher scaffolding (measured in intermediate tokens emitted at inference) does masked distillation need to match the non-masked variant on OOD, while still keeping a large improvement in inference efficiency?
- **RQ3.** Masked distillation reduces inference cost at the price of additional training-time cost. Is this a favourable amortization tradeoff, do the inference-time savings outweigh the extra training cost?
- **RQ4.** Does the training objective matter? Specifically, does on-policy reverse-KL distillation, with its mode-seeking behaviour and reduced exposure bias, lead to different scaffolding–generalization tradeoffs than supervised fine-tuning on teacher rollouts?

Our experiments on GSM8K and Countdown show that the answer depends strongly on the dataset. On GSM8K, the masked and non-masked students reach similar accuracy in-distribution (45.26% vs. 54.36%) and the same accuracy on the two OOD splits (0% on AIME-25 and 10.60% on MATH-500). The masked student produces roughly $5\times$ fewer tokens at inference, at no measurable cost to OOD accuracy. On Countdown the gap between the two is much larger. The fully masked student reaches 34.08% in-distribution, while non-masked reaches 81.05% and the teacher reaches 87.30%. Adding 100 or $1{,}000$ teaching tokens recovers a large part of this gap (67.25% and 65.04% respectively). The same ordering holds on the OOD splits at lower absolute levels: under a target-range shift the masked variants reach 15.04%, 44.25%, and 50.74% against 76.84% for non-masked; under a search-depth shift the masked variants collapse to near zero while non-masked still reaches 9.20% on 5-input problems. These results suggest that the cost of fully internalizing the teacher's intermediate tokens depends on the structure of the task: on simpler arithmetic the cost is small and the inference savings come essentially for free, while on Countdown the cost is large and only partially recovered by teacher-token scaffolding.

The rest of the paper is organized as follows. Section 2 gives a 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 $L_{\text{SD}}(\theta)$ over the dataset (X, Y) is calculated as

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

- **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.

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

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 smaller, 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 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, $\pi^T(\cdot \mid x, \text{ITs})$.

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; we describe the instruction prompt and the teaching-token scaffolding used in this phase in the next subsection. 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 $\pi^T(\cdot \mid x, \text{ITs})$, 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

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

where $\mathcal{V}$ is the vocabulary. Since we use same family of model as teacher and student the vocabulary of both the models are same.

We compare masked distillation against a *non-masked* variant in which both the teacher and the student are conditioned

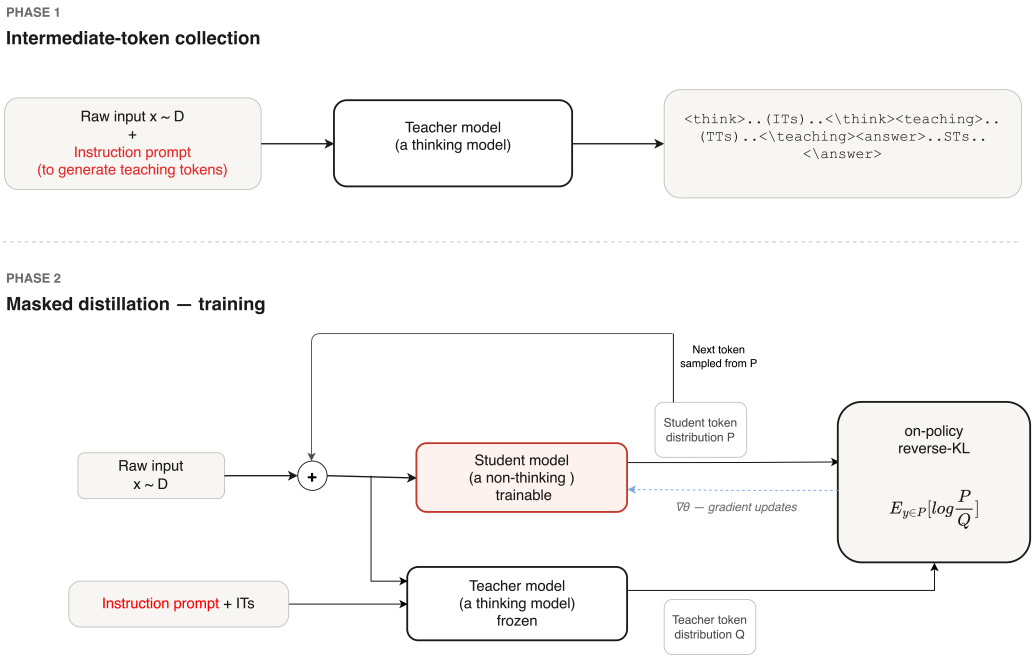

Figure 1. **Overview of the masked distillation framework.** *Phase 1 intermediate-token collection (top):* for each input question $x \sim \mathcal{D}$, we sample a response from the thinking teacher π^T together with an instruction prompt that asks the teacher to emit a concise scaffold. The response has the form `<think>(ITs)</think><teaching>(TTs)</teaching><answer>(STs)</answer>`, from which we extract the intermediate tokens (ITs), teaching tokens (TTs), and 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, the instruction prompt, and the ITs collected in Phase 1. 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.

on the same prompt (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

$$\mathcal{L}_{\text{NMKD}} = D_{\text{KL}}\Big(\pi_\theta^S(\cdot \mid x, y_{<t}) \,\Big\|\, \pi^T(\cdot \mid x, y_{<t})\Big). \quad (5)$$

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 two intermediate variants, *masked-k* for $k \in \{100, 1000\}$—in which the student is trained to emit k *teaching tokens* before the solution.

What are these 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.

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:

$$\mathcal{L}_{\text{SFT}} = -\,\mathbb{E}_{x\sim\mathcal{D},\; y\sim\pi^T(\cdot\mid x,\,\text{ITs})}\left[\sum_{t=1}^{|y|} m_t \,\log \pi_\theta^S(y_t \mid x, y_{<t})\right], \quad (6)$$

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).

4. Experimental Setup

Datasets: We present our analysis on two datasets - 1) the *GSM8K* dataset (Cobbe et al., 2021), which is a widely used benchmark dataset consisting of grade school math problems, 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. And, 2) *Countdown* dataset (Countdown.), which 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 this 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.

Models: For training on the GSM8K dataset, we use Qwen2.5-0.5B-Instruct and Qwen3-1.7B as the student and teacher models, respectively. For training on the Countdown dataset, we use Qwen2.5-0.5B-Instruct and Qwen3-4B as the student and teacher models, respectively.

Technical Setup and Hyperparameters: 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). The student is Qwen2.5-0.5B-Instruct in all experiments; the teacher is Qwen3-1.7B for GSM8K and Qwen3-4B for Countdown. 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-$p = 0.95$. The masked-distillation loss is the reverse-KL divergence between the student and teacher next-token distributions, computed over the full vocabulary. We optimize with AdamW at a constant learning rate of 1×10^{-5}.

Table 1. **Test accuracy on GSM8K and OOD math benchmarks.** The standard GSM8K test set is the in-distribution split; AIME-25 (30 problems) and MATH-500 (500 problems) are OOD splits that share the arithmetic substrate but exercise it at substantially higher difficulty. The *N examples* row gives the size of each split.

Methods *N examples*	**GSM8K** (1319)	**AIME-25** (30)	**MATH-500** (500)
Base student	15.69%	0.00%	8.80%
Teacher (1.7B)	89.23%	23.33%	83.60%
MD	45.26%	0.00%	10.60%
Non-MD	54.36%	0.00%	10.60%

5. Results and Discussion

Tables 1–4 report final accuracy and average response-token length on GSM8K and Countdown, across in-distribution (ID) and out-of-distribution (OOD) splits, for all methods we evaluate; these tables address RQ1 (the OOD cost of full internalization) and RQ2 (the scaffolding budget required to recover OOD performance). We additionally analyze when the inference-time savings of masked distillation amortize the extra training-time compute (RQ3), and compare on-policy reverse-KL distillation with supervised fine-tuning on teacher rollouts (RQ4).

Table 2. **Mean response length (output tokens) on GSM8K and OOD math benchmarks.** Each cell shows *pos*/*neg*: *pos* (left) is the mean length over correct responses; *neg* (right) is the mean length over incorrect responses. A dash (-) indicates an empty bucket.

Methods	**GSM8K**	**AIME-25**	**MATH-500**
Base student	113.9 / 289.0	- / 2609.3	217.8 / 943.8
Teacher (1.7B)	2092.7 / 4947.6	6137.4 / 8156.5	3607.9 / 7739.9
MD	505.4 / 1685.1	- / 4825.4	802.6 / 2677.9
Non-MD	2689.6 / 6208.0	- / 8044.2	4952.0 / 7585.1

Evaluation splits. For **GSM8K** we evaluate on the standard test set (1,319 problems) as ID, and use AIME-25 (30 problems) and MATH-500 (500 problems) as OOD splits that share the arithmetic substrate but exercise it at substantially higher difficulty. For **Countdown**, the ID test set matches the training distribution (3-4 two-digit input numbers, two-digit target; 1,024 problems), and the three OOD splits each perturb a single 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). The base student is at 0% on every Countdown split that means all signal in the Countdown evaluation comes from the distillation it-

Table 3. **Test acccuracy 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, two-digit target). The *N examples* row gives the size of each split.

	In-Distribution	**Out-of-Distribution**		
Methods	(3–4 nums, 2d tgt)	(3–4 nums, 1d tgt)	(5 nums)	(6 nums)
N examples	(1024)	(678)	(522)	(478)
Base student	0.00%	0.00%	0.00%	0.00%
Teacher (4B)	87.30%	83.48%	25.67%	18.83%
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%

Table 4. **Mean response length (output tokens) on Countdown.** Each cell shows *pos*/*neg*: *pos* (left) is the mean length over correct responses; *neg* (right) is the mean length over incorrect responses. A dash (-) indicates an empty bucket.

	In-Distribution	**Out-of-Distribution**		
Experiment	(3–4 nums, 2d tgt)	(3–4 nums, 1d tgt)	(5 nums)	(6 nums)
Base student	- / 88.0	- / 13.6	- / 16.2	- / 16.4
Teacher (4B)	1715.9 / 7904.0	1309.5 / 7704.2	2964.6 / 7911.6	4191.0 / 7992.8
MD	171.5 / 524.0	372.1 / 499.7	- / 469.4	- / 872.1
MD (100 TTs)	173.9 / 5993.2	265.5 / 6578.5	260.0 / 6680.3	- / 5853.8
MD (1k TTs)	766.5 / 6284.2	729.6 / 5733.6	872.0 / 5540.6	- / 5524.1
Non-MD	1982.0 / 7633.3	1761.1 / 7110.4	5046.9 / 6850.7	- / 6584.8

self, whereas on GSM8K the base student is already at 15.69%, since Qwen2.5-0.5B-Instruct retains some grade-school arithmetic ability before distillation.

In-distribution behaviour: the cost of full internalization. The in-distribution cost of asking the student to answer directly is sharply task-dependent. On GSM8K (Table 1), the non-masked distillation trained student (Non-MD) reaches 54.36% and the fully masked distillation trained student (MD) reaches 45.26%, a gap of only ∼9 points; both lag the 1.7B teacher (89.23%). On Countdown (Table 3), Non-MD reaches 81.05%—∼6 points below the 4B teacher's 87.30%, while MD reaches only 34.08%, a 47-point gap. The main observation is that the in-distribution penalty for full internalization is small on GSM8K and large on Countdown, and this ID gap sets a floor against which the OOD numbers should be read.

Out-of-distribution behaviour (RQ1). On the GSM8K OOD splits both MD and Non-MD collapse equally: both reach 0% on AIME-25 and 10.60% on MATH-500, against teacher accuracies of 23.33% and 83.60% respectively. The differences between MD and Non-MD on GSM8K OOD are within noise, both methods fail to transfer in the same way suggesting that the in-distribution gap on GSM8K is not the result of MD failing to absorb computation that Non-MD preserves; rather, neither distilled student is able to recover the teacher's OOD performance regardless of scaffolding choice. On Countdown the picture is qualitatively different: the in-distribution gap widens sharply under distribution shift. On OOD-1 (target-range shift), Non-MD remains close to the teacher (76.84% vs. 83.48%) while MD drops to 15.04%. On OOD-2 (5 input numbers), where the teacher itself drops to 25.67%, MD collapses to 0% while Non-MD retains 9.20%, about a third of the teacher's OOD ceiling. On OOD-3 (6 input numbers), the teacher is at 18.83% but every distilled variant, including Non-MD, collapses to 0%, indicating that this split exceeds the OOD reach of all distilled students. The asymmetry between Non-MD and MD on Countdown is therefore not an additive offset: full internalization loses disproportionately more on OOD than on ID, and the gap grows with the magnitude of the search-depth shift. Together, the two domains give a clean answer to **RQ1**: full masked distillation pays a disproportionate OOD penalty on tasks with non-trivial search structure (Countdown), while paying essentially no extra

OOD penalty on simpler tasks where the OOD ceiling is already low for all distilled students (GSM8K).

Effect of teacher-token scaffolding (RQ2). We sweep the teacher-token scaffolding budget $k \in \{0, 100, 1\text{k}\}$ on Countdown, where the MD vs Non-MD gap is largest and the scaffolding axis is therefore informative. We do not run the same sweep on GSM8K, since the MD vs Non-MD ID gap is already small ($\sim$9 points) and both methods collapse identically OOD, leaving no slack for scaffolding to recover. On the Countdown ID split, MD with 100 and 1k teaching tokens recovers from 34.08% to 67.25% and 65.04% respectively, roughly double the fully masked accuracy and within $\sim$15 points of Non-MD. The recovery is even more pronounced on OOD-1: the masked variants reach 44.25% (100 TTs) and 50.74% (1k TTs), against 15.04% for fully masked, showing that even a small teacher-token scaffold restores most of the OOD performance lost under full masking, and that the trend is monotone in the scaffolding budget (1k TTs $>$ 100 TTs $>$ 0 TTs) on both ID and OOD-1. On the harder search-depth splits (OOD-2 and OOD-3), neither 100 nor 1k teaching tokens are sufficient to lift the student off the 0% floor, while Non-MD still preserves a non-trivial 9.20% on OOD-2. The scaffolding axis therefore recovers OOD performance under target-range shift but not under search-depth shift, a partial answer to **RQ2**: there exists a scaffolding budget at which masked distillation matches Non-MD on mild OOD shifts, but no budget within our $\{0, 100, 1\text{k}\}$ sweep matches Non-MD under length shift.

Inference cost. Tables 2 and 4 confirm that the accuracy comparisons translate directly to large differences in inference cost. On GSM8K, MD emits 505.4 tokens per correct response on average, against 2689.6 for Non-MD and 2092.7 for the teacher roughly a $5\times$ reduction at essentially the same OOD performance. On Countdown ID, the gap is even larger: MD emits 171.5 tokens per correct response, against 1982.0 for Non-MD and 1715.9 for the teacher, a 10–12$\times$ reduction. The 100-TT and 1k-TT variants on Countdown sit between these extremes (173.9 and 766.5 correct-response tokens respectively), as expected from their scaffold budgets. Combined with the accuracy results, this gives a clean picture of the scaffolding–generalization trade-off: under target-range shift on Countdown, increasing the teaching-token budget improves OOD accuracy at a roughly proportional cost in response length, and the operating point can be chosen to fit a given inference-cost budget; under search-depth shift, no setting in our sweep is on the Pareto frontier with the non-masked baseline, suggesting that fully internalizing the teacher's reasoning is insufficient when the student must extrapolate beyond the training search depth.

Training-time vs. inference-time cost (RQ3). Figure 2 shows validation accuracy on Countdown as a function of training step for the four variants. The non-masked baseline converges fastest, reaching $\sim$0.80 by step 200 and plateauing near 0.81; this is consistent with the student receiving token-level supervision over the teacher's full think-then-answer trace. The masked variants converge more slowly: $k = 100$ and $k = 1{,}000$ teaching tokens climb gradually to $\sim$0.65 and $\sim$0.70 by step 1,100 and are still trending upward at the end of training, while fully masked distillation plateaus near 0.34.

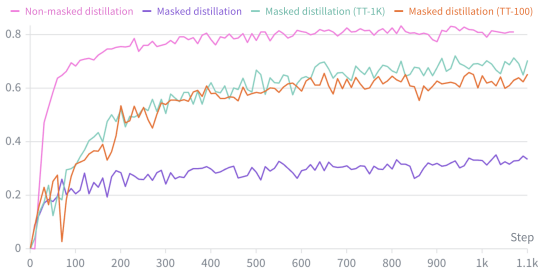

Figure 2. Countdown validation accuracy over training steps

All variants in our experiments share the same training set and a similar number of optimization steps. This is a conservative budget for the masked variants: their per-question supervision is concentrated on the solution tokens, whereas non-masked supervision is dense across the entire trace, so the masked variants have strictly fewer effective gradient signals per question. A longer or larger-data training schedule may therefore close some of the in-distribution gap we report in Section 5. We do not run a controlled GPU-hour comparison and therefore do not attempt a precise amortization break-even calculation. The directional evidence is nonetheless clear: per-query inference cost is reduced by $\sim 11\times$ (fully masked and $k = 100$) and $\sim 2.6\times$ ($k = 1{,}000$) relative to non-masked on Countdown (Table 4), while per-step training cost is similar across variants (the masked variants pay only a small overhead for the longer teacher prompt that includes the intermediate tokens). Under any realistic deployment volume this places masked distillation in a favourable amortization regime; a precise GPU-hour break-even calculation, together with a training schedule that matches effective supervision across variants, is left to future work.

Effect of training objective (RQ4). Table 5 reports the same scaffolding sweep under supervised fine-tuning on teacher rollouts (Masked-SFT and Non-masked SFT), in place of the on-policy reverse-KL objective used in Table 3.

Two patterns emerge - First, *teacher-token scaffolding does not help under SFT*. The fully masked SFT student reaches 38.4% on the in-distribution split; the 100- and 1,000-TT variants reach 38.5% and 30.8% respectively, essentially flat, and in the 1k case worse than fully masked. 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%). The same flat pattern holds on the target-range OOD split, where the three Masked-SFT variants reach $\sim$10–12% against 15 / 44 / 51% under reverse-KL, and on the search-depth splits, where every Masked-SFT variant collapses to 0%.

Second, *the methodology gap is concentrated on the partial scaffold variants*, not on the no-scaffold or full-trace endpoints. In the fully masked setting (no teaching tokens), the two objectives are within a few points of each other (38.4% for SFT vs. 34.08% for reverse-KL). At the other endpoint, Non-masked SFT reaches 75.7% in-distribution and 69.3% on target-range OOD, only 5–8 points below Non-MD's 81.05% / 76.84% under reverse-KL. The two methodologies separate sharply only once the student is asked to use a partial scaffold: SFT does not learn to exploit the 100- or $1{,}000$-TT scaffold, while reverse-KL does.

Table 5. Countdown test accuracy on, in and out distribution data split for SFT variant.

	In-Distribution	**Out-of-Distribution**		
Methods	(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%

Because under SFT, the student never generates its own teaching tokens during training, so when it produces them at inference the conditioning prefix on which it must produce the answer is itself out-of-distribution from training. Under on-policy reverse-KL, the student is supervised on its own generations, so the partial scaffold it emits at inference is the one it has been optimised to condition on. Together with the scaffolding-sweep results reported in Table 3, this answers **RQ4**: the training objective matters substantially, but its effect is not uniform across the scaffolding axis. The two objectives behave similarly at the endpoints (no scaffold, full trace) and diverge sharply in the middle, where reverse-KL on student-sampled tokens leverages teacher-token scaffolding while supervised fine-tuning on teacher rollouts does not. The choice of training objective and the choice of scaffolding budget are therefore not independent design variables, and any cost–benefit account of internalisation must specify both.

6. Conclusion

In this paper we study how much of a thinking teacher's intermediate-token computation can be internalized into a smaller, non-thinking student through distillation. We proposed **masked distillation**, which supervises the student only on solution tokens while masking out the teacher's intermediate tokens at training time, and compared it against a non-masked baseline that reproduces the full thinking trace at inference. By varying the teacher scaffolding budget ($k \in \{100, 1000\}$ teaching tokens) and the training objective (on-policy reverse-KL distillation vs supervised fine-tuning on teacher rollouts), we mapped how the inference-cost savings of internalization trade off against in and out-of-distribution accuracy on GSM8K and Countdown.

Across both domains we find that the cost of full internalization depends strongly on the task (or can also be a function of what data contamination the pre-trained model already have). On GSM8K, masked and non-masked students reach similar in-distribution accuracy and identical OOD accuracy on AIME-25 and MATH-500, while the masked student produces about $5\times$ fewer tokens at inference. On Countdown the gap is much larger, and even $1{,}000$ teaching tokens of scaffolding only partially close it; under search-depth OOD shift every distilled variant collapses. The masked variants are trained longer than non-masked but emit roughly $11\times$ fewer tokens per query at inference (about $2.6\times$ for the 1k-TT variant), so the inference-time savings amortize the additional training cost. The training objective matters substantially: SFT and reverse-KL behave similarly at the no-scaffold and full-trace endpoints but diverge sharply at partial scaffolding, with SFT students failing to use teacher-token scaffolding while reverse-KL students leverage it effectively. The broader takeaway is that in-distribution evaluations of internalization methods can be misleading: a student that looks competitive on the training distribution may collapse under modest distribution shift, and the size of this collapse depends on both the scaffolding budget and the choice of training objective.

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, our scaffolding sweep stops at $k = 1000$, and no scaffold within this range recovers the non-masked performance under search-depth shift. Whether richer scaffolds can close this remaining gap remains an open question. 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.

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.

Cheng, J. and Van Durme, B. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*, 2024.

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.

Geiping, J., Yang, X., and Su, G. Efficient parallel samplers for recurrent-depth models and their connection to diffusion language models. *arXiv preprint arXiv:2510.14961*, 2025.

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.

Hao, S., Sukhbaatar, S., Su, D., Li, X., Hu, Z., Weston, J., and Tian, Y. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

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

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.

London, C. and Kanade, V. Pause tokens strictly increase the expressivity of constant-depth transformers. *arXiv preprint arXiv:2505.21024*, 2025.

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.

Merrill, W. and Sabharwal, A. The expressive power of transformers with chain of thought. *arXiv preprint arXiv:2310.07923*, 2023.

Pfau, J., Merrill, W., and Bowman, S. R. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*, 2024.

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.

Shen, Z., Yan, H., Zhang, L., Hu, Z., Du, Y., and He, Y. CODI: Compressing chain-of-thought into continuous space via self-distillation. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

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.

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

Su, D., Zhu, H., Xu, Y., Jiao, J., Tian, Y., and Zheng, Q. Token assorted: Mixing latent and text tokens for improved language model reasoning. *arXiv preprint arXiv:2502.03275*, 2025.

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.

Wen, X., Huang, J., Li, Z., Li, M., Zhong, J., Xu, Z., Yuan, M., Huang, Y., and Xu, Q. Reasoning scaffolding: Distilling the flow of thought from llms. *arXiv preprint arXiv:2509.23619*, 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.

Zhao, C., Tan, Z., Ma, P., Li, D., Jiang, B., Wang, Y., Yang, Y., and Liu, H. Is chain-of-thought reasoning of llms a mirage? a data distribution lens. *arXiv preprint arXiv:2508.01191*, 2025.

A. 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 6 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 6. **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	∼100 words	80	73	50–99 words
$k = 1{,}000$ TTs	∼1,000 words	389	383	300–499 words