SMC-KD: KNOWLEDGE DISTILLATION USING TEACHER-GUIDED SEQUENTIAL MONTE CARLO

Anonymous authors
Paper under double-blind review

ABSTRACT

Small language models are becoming more capable, and in many domains they can generate a correct answer across multiple samples, even when they do not do so reliably in a single attempt. Distillation from a stronger teacher can turn these occasional successes into more reliable behavior, but its effectiveness depends on which responses the student is trained on. For example, fine-tuning on teacher-generated responses can lead to exposure bias: the student trains on prefixes it would not produce itself, which differ from those it sees at inference. Training on the student's own responses avoids this mismatch but needs feedback on those responses. Selecting good responses with Best-of-N (BoN) or verifier-guided search typically relies on correctness labels or separately trained reward models. On-policy distillation (OPD) instead has the teacher score each student token, but a poorly matched teacher can hinder performance, and different teacher and student vocabularies complicate token-level supervision. In this paper, we introduce teacher-guided Sequential Monte Carlo Knowledge Distillation (SMC-KD), a method that uses teacher feedback to identify desirable trajectories among those generated by the student and distills them into the student's parameters. SMC-KD requires no correctness labels, process reward model, or semantic step boundaries, and naturally supports teacher and student models with different vocabularies. The student generates a population of partial trajectories, the teacher scores them by the teacher–student likelihood ratio, and SMC resampling allocates subsequent generation to higher-scoring prefixes. We then train the student on these trajectories through supervised fine-tuning (SFT). Before any training, we show that teacher-guided SMC selects more accurate responses than BoN under the same likelihood score, for every student–teacher pairing and population size we evaluate. After training, SMC-KD matches or improves on the base student in every setting and outperforms OPD in 10 of 12 comparisons across four student–teacher pairings and three math benchmarks.

1 INTRODUCTION

Small language models are becoming increasingly capable at reasoning (e.g., Qwen et al., 2025; Yang et al., 2025; Team et al., 2025; Grattafiori et al., 2024), and on many reasoning tasks they can find a correct answer across several samples, even when they do not do so reliably in a single attempt. Figure 1 illustrates this gap for four small models on MATH-500: the probability that at least one of eight samples is correct (pass@8) exceeds single-sample accuracy (pass@1) by 22 to 29 percentage points. Therefore, correct solutions are within these models' reach. Brown et al. (2024) show the same pattern across other models and tasks, and Chen et al. (2026) find that much of the gain from reinforcement learning on reasoning tasks comes from making such reachable solutions more likely. Another way to turn these occasional successes into reliable behavior is knowledge distillation (KD), which transfers capabilities from a stronger teacher into a smaller student (Hinton et al., 2015; Xu et al., 2024). How well it does so depends on which responses the student is trained on.

A common approach fine-tunes the student on responses generated by the teacher (Kim & Rush, 2016; Ho et al., 2023; Hsieh et al., 2023). During this supervised fine-tuning (SFT), the student predicts each token after a teacher-written prefix. At inference, however, it continues its own prefixes instead, and an early mistake can lead it into contexts that never appeared in training. This

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

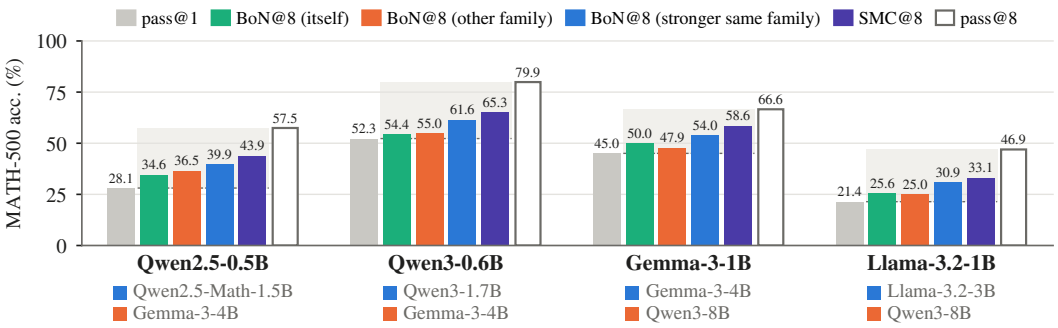


Figure 1: **Student model accuracy on MATH-500 under different sampling and selection rules.** For each student, we report pass@1, pass@8, and Best-of-8 (BoN@8) when scoring completed responses with the student itself, a stronger model from another family, or a stronger model from the same family. We also report SMC@8 using eight particles and the same scorer as the same-family BoN@8. The gap between pass@1 and pass@8 indicates that correct responses are available through repeated sampling but are not generated reliably in a single attempt. Scoring and selection recovers part of this gap, with teacher-guided SMC achieving the highest selected-response accuracy across all settings. Complete results with multi-run statistics are included in Table 4 and Table 5.

mismatch between training and inference prefixes is known as exposure bias (Bengio et al., 2015; Ranzato et al., 2016). On-policy distillation (OPD) avoids this mismatch by training on responses sampled from the student (Agarwal et al., 2024; Gu et al., 2024). At each student-written prefix, the teacher's next-token distribution indicates which tokens the student should prefer, so the student learns from its own mistakes. This feedback is only as good as the teacher's predictions on student-written text. On prefixes that the teacher would rarely produce, its token-level feedback can become unreliable (Wang et al., 2026; Fu et al., 2026), and a large divergence between teacher and student can destabilize training (Liu et al., 2026). Token-level supervision also assumes a shared vocabulary; otherwise, the token distributions of the two models must first be aligned (Boizard et al., 2025; Minixhofer et al., 2026).

The gap between pass@1 and pass@8 suggests another use of the student's own responses: rather than correcting its mistakes, we can find its successes and train on them. Because these responses come from the student, training on them also avoids the teacher-prefix mismatch. This is itself a form of distillation, in which the rule that selects responses plays the role of the teacher. The difficulty lies in finding good responses, and existing methods rely on correctness labels or a separately trained model to do so. Self-training keeps sampled solutions whose final answers match known labels (Zelikman et al., 2022; Yuan et al., 2023; Singh et al., 2024), Best-of-N (BoN) sampling ranks complete responses with a trained verifier (Cobbe et al., 2021), process reward models (PRMs) score intermediate steps to guide search (Lightman et al., 2024; Wang et al., 2024; Snell et al., 2024), and Sequential Monte Carlo (SMC) methods reweight and resample partial responses using PRM scores or learned value functions (Puri et al., 2026; Feng et al., 2025).

In this work, we investigate a label-free alternative: can a stronger teacher's likelihood identify good student responses? Figure 1 answers this with BoN over eight complete responses. Selecting by the student's own likelihood already improves on pass@1 for every student, though only modestly. Selecting by the teacher–student likelihood ratio, with a stronger teacher from the same family, improves further for every student. A stronger model from another family does not consistently improve on self-scoring, which suggests that a stronger scorer helps most when it is closely matched to the student (Appendix D repeats this comparison for every generator–scorer pair among 13 models). The teacher's likelihood is therefore a useful signal, but BoN applies it only after each response is complete. SMC can apply a score throughout generation: it maintains a population of partial responses and resamples them according to incremental weights (Lew et al., 2023). With the teacher–student likelihood ratio as the score, the student extends its partial responses block by block, and resampling gives higher-scoring prefixes more continuations. With the same teacher and eight particles, SMC achieves the highest accuracy for every student in Figure 1. Therefore, teacher-

guided SMC finds better student responses than BoN without any correctness labels, which makes it a natural source of training data.

Building on these observations, we introduce **teacher-guided Sequential Monte Carlo Knowledge Distillation (SMC-KD)**, which uses teacher feedback to identify desirable trajectories among those generated by the student and distills them into the student's parameters. For each prompt, the student generates a population of partial responses in blocks of tokens. The teacher scores each new block by the teacher–student likelihood ratio, and SMC resampling allocates subsequent generation to higher-scoring prefixes. Copies of a resampled prefix are extended independently, so a promising prefix can lead to several different continuations. As a result, every token of the final responses is generated by the student and the teacher only decides which prefixes receive more generation. We then train the student on the final population through SFT, and can repeat this process with the updated student. SMC-KD requires no correctness labels, process reward model, or semantic step boundaries, and since the teacher scores text rather than supervising individual tokens, the teacher and student can use different vocabularies. After training, the student generates responses on its own, without the teacher or particle sampling. We also formulate SMC-KD as KL-regularized policy improvement followed by an SFT projection (Section 3.3), which shows that with the likelihood-ratio score, the improvement step shares its objective with reverse-KL OPD, while the SFT projection minimizes the forward rather than the reverse KL divergence.

Our experiments first evaluate teacher-guided SMC as a response selector and then evaluate SMC-KD as a distillation method. Before any training, we show that teacher-guided SMC selects more accurate responses on MATH-500 than BoN under the same likelihood-ratio score (Section 4.2). This holds for each of four student–teacher pairings from the Qwen, Gemma, and Llama families and for every population size from 4 to 32. After training, SMC-KD matches or improves on the base student in every setting and outperforms OPD in 10 of 12 comparisons across four Qwen and Llama pairings and three benchmarks: MATH-500, GSM8K, and AMC23 (Section 4.3). We also show that SMC-KD is more robust than OPD to student–teacher mismatch: when the Qwen2.5-0.5B student is distilled from Qwen3-1.7B instead of Qwen2.5-Math-1.5B, OPD accuracy drops on all three benchmarks, while SMC-KD drops less and remains above OPD on each.

2 PRELIMINARIES

We first introduce the notation used throughout the paper, then review the knowledge distillation objectives and the Sequential Monte Carlo (SMC) framework underlying our method. A detailed discussion of related work is provided in Appendix A.

2.1 NOTATION

Let q denote an input prompt, $\pi_S = \pi_\theta$ the student policy, and π_T a fixed teacher policy. The student generates a response in blocks of C tokens, with the final block potentially shorter if generation terminates early. Blocks are defined by token counts and do not need to align with semantic reasoning steps. Generation stops when an end-of-sequence token is produced or after at most H blocks. A response that has ended is extended by an empty block, so it remains unchanged in later rounds. Let y_h denote the block generated at round h. After h rounds, the partial response is $y_{1:h} = (y_1, \ldots, y_h)$ and the corresponding prompt–response prefix is $s_h = [q, y_{1:h}]$, where $[\cdot, \cdot]$ denotes concatenation and $s_0 = q$. We use Y to denote the complete response and $|Y| \leq H$ for its number of blocks. Section 2.3 reviews how SMC maintains a population of N such prefixes, called particles, and reweights and resamples them between rounds.

2.2 KNOWLEDGE DISTILLATION

Knowledge distillation trains a student model to imitate the output distribution of a more capable teacher (Hinton et al., 2015). At the sequence level, a widely used objective is the forward KL divergence:

$$D_{\mathrm{KL}}(\pi_T(\cdot \mid q) \,\|\, \pi_S(\cdot \mid q)) = \text{const.} - \mathbb{E}_{Y \sim \pi_T(\cdot|q)} \left[\log \pi_S(Y \mid q)\right], \tag{1}$$

where the constant is independent of the student parameters. Minimizing equation 1 is equivalent to maximizing the student's likelihood of teacher-sampled responses, which can be implemented

through supervised fine-tuning (Kim & Rush, 2016). When trained this way, the student conditions on teacher-generated prefixes, but at inference it conditions on its own output. This mismatch, known as *exposure bias* (Bengio et al., 2015; Ranzato et al., 2016), means that an early student error can move later generation into contexts that were not represented during training.

To address this, on-policy distillation trains on student-generated trajectories using teacher feedback (Agarwal et al., 2024; Gu et al., 2024), often through the reverse KL divergence:

$$D_{\mathrm{KL}}(\pi_S(\cdot \mid q) \,\|\, \pi_T(\cdot \mid q)) = \mathbb{E}_{Y \sim \pi_S(\cdot|q)} \left[\log \frac{\pi_S(Y \mid q)}{\pi_T(Y \mid q)} \right]. \tag{2}$$

Unlike the forward-KL objective above, the expectation is now taken over student-generated responses, allowing the teacher to provide feedback on the student's own trajectories. Practical on-policy distillation methods commonly query the teacher's next-token distribution at each student prefix and use it as a dense token-level training signal (Agarwal et al., 2024; Gu et al., 2024).

However, student-generated prefixes may be out of distribution for the teacher, which makes its token-level feedback on them unreliable (Fu et al., 2026). Wang et al. (2026) call this problem *reversed exposure bias* because it mirrors the exposure bias of SFT, with the teacher rather than the student now conditioning on prefixes it would not produce itself. Which trajectories the student trains on therefore matters as much as the distillation objective. In our work, the teacher helps choose these trajectories instead of correcting them token by token. It scores blocks of student-generated text, and the student is then trained on the resulting trajectories through SFT. Because the teacher never supervises individual tokens, the two models also need not share a vocabulary, which token-level feedback requires unless their token distributions are first aligned (Boizard et al., 2025; Minixhofer et al., 2026). Section 3.3 shows that, with the teacher–student likelihood-ratio score, this procedure shares its improvement objective with reverse-KL OPD equation 2, while its SFT projection minimizes the forward KL divergence equation 1. Generating these trajectories requires sampling from a distribution that reweights the student's responses toward those the teacher scores highly, and the next subsection reviews SMC, the sampling method we use for this purpose.

2.3 Sequential Monte Carlo

SMC is a method for sampling from a target distribution that is hard to sample directly. It evolves a population of weighted samples, called particles (Lew et al., 2023; Zhao et al., 2024). For language models, each particle is a partial response. In each round, SMC extends every particle by one block and reweights it by how much the new block changes its score. It then resamples the population, copying high-weight prefixes and dropping low-weight ones. We now formalize this procedure following Golowich et al. (2026). Let π_{ref} be a proposal policy and let $r^*(Y \mid q) \geq 0$ be a reward on complete responses. Together, they define the target distribution

$$\pi_H^*(Y \mid q) = \frac{\pi_{\mathrm{ref}}(Y \mid q) r^*(Y \mid q)}{Z^*(q)}, \tag{3}$$

where $Z^*(q) = \mathbb{E}_{Y \sim \pi_{\mathrm{ref}}}[r^*(Y \mid q)]$. For example, in the math domain, $r^*(Y \mid q)$ can be 1 if the final answer is correct and 0 otherwise, which makes π_H^* the proposal conditioned on having a correct answer.

The target is defined on complete responses, but π_{ref} produces them one block at a time, so their probability factorizes as $\pi_{\mathrm{ref}}(Y \mid q) = \prod_{k=1}^{|Y|} \pi_{\mathrm{ref}}(y_k \mid s_{k-1})$. Steering generation toward π_H^* therefore requires a score for partial responses. For a prefix $s_h = [q, y_{1:h}]$, the natural score is the exact continuation value $V^*(s_h) = \mathbb{E}_{y_{h+1:H} \sim \pi_{\mathrm{ref}}(\cdot|s_h)}[r^*(Y \mid q)]$, which equals r^* when the response is complete. In principle, one could sample from π_H^* block by block, drawing each block from $\pi_{\mathrm{ref}}(y_h \mid s_{h-1}) V^*(s_h) / V^*(s_{h-1})$. Nevertheless, this is rarely possible in practice, because V^* averages the reward over all continuations and is generally intractable.

In SMC, V^* is therefore approximated with a nonnegative potential $\widehat{V}$ that satisfies $\widehat{V}([q, Y]) = r^*(Y \mid q)$ on complete responses. However, $\widehat{V}$ cannot simply replace V^* in the block-by-block sampler. The exact sampler sums to one over y_h because $V^*(s_{h-1}) = \mathbb{E}_{y_h \sim \pi_{\mathrm{ref}}(\cdot|s_{h-1})}[V^*(s_h)]$. The potential $\widehat{V}$ generally lacks this property, so $\pi_{\mathrm{ref}}(y_h \mid s_{h-1}) \widehat{V}(s_h) / \widehat{V}(s_{h-1})$ need not sum to one. SMC instead samples the next block of each of N partial responses from π_{ref}, weights each

extension by $\widehat{V}(s_h)/\widehat{V}(s_{h-1})$, and normalizes the weights across the population. These weights steer the population toward the intermediate target distributions $\widehat{\pi}_h(y_{1:h} \mid q) \propto \pi_{\mathrm{ref}}(y_{1:h} \mid q)\widehat{V}(s_h)$, which replace V^* with $\widehat{V}$ in the ideal prefix distributions $\pi^*_h(y_{1:h} \mid q) \propto \pi_{\mathrm{ref}}(y_{1:h} \mid q)V^*(s_h)$. Since $\widehat{V} = r^*$ on complete responses, the final target $\widehat{\pi}_H$ equals π^*_H. Notably, the intermediate targets are only as accurate as $\widehat{V}$. A good potential gives high values to prefixes that are likely to lead to high-reward responses, so SMC spends more of its generation on those prefixes. Algorithm 1 gives the resulting SMC procedure.

The algorithm alternates between resampling and extension. At each round $h = 1, \ldots, H$, it draws N ancestors from the previous weighted population $\widehat{\nu}_{h-1}$ and extends each selected prefix independently by one block from π_{ref}. The incremental ratio $\widehat{V}(s_h)/\widehat{V}(s_{h-1})$ then accounts for the change in potential caused by the new block. Particles whose responses have already ended are carried forward with incremental weight one, so the population always contains N particles and finished responses can still be resampled. Resampling gives high-weight prefixes more opportunities to produce continuations, and copies of the same prefix are extended independently and can branch. Generally, standard resampling schemes include multinomial, stratified, and systematic resampling (Douc & Cappé, 2005). In this work, we use systematic resampling, applied only when the effective sample size (ESS) of the weights falls below $N/2$ (Appendix C). After the last round, a response drawn from the final weighted population $\widehat{\nu}_H$ follows a distribution that approximates π^*_H. Under coverage conditions, the approximation error decreases as N grows and as each $\widehat{\pi}_h$ approaches π^*_h (Golowich et al., 2026).

Algorithm 1 Sequential Monte Carlo

Require: Prompt q, proposal π_{ref}, potential $\widehat{V}$, particle count N, rounds H

$\widehat{\nu}_0 \leftarrow \delta_q$ ▷ point mass at q
for $h = 1, \ldots, H$ **do**
  $\widetilde{s}^{1:N}_{h-1} \leftarrow \textsc{Resample}(\widehat{\nu}_{h-1}, N)$
  for $i = 1, \ldots, N$ **do**
    $y^i_h \sim \pi_{\mathrm{ref}}(\cdot \mid \widetilde{s}^i_{h-1})$
    $s^i_h \leftarrow [\widetilde{s}^i_{h-1}, y^i_h]$
    $w^i_h \leftarrow \widehat{V}(s^i_h)/\widehat{V}(\widetilde{s}^i_{h-1})$
  end for
  $W_h \leftarrow \sum_{i=1}^N w^i_h$
  $\widehat{\nu}_h \leftarrow \sum_{i=1}^N (w^i_h/W_h)\, \delta_{s^i_h}$
end for
return $\widehat{\nu}_H$

3 Method: SMC-KD

As noted in Section 2.3, the intermediate targets of SMC are only as accurate as the potential $\widehat{V}$. For a general reward, finding a good potential is difficult, because V^* is intractable and approximating it usually requires a process reward model or a value function trained with correctness labels. In distillation, however, the teacher offers reasonable choices without such training, because its likelihood can already score partial responses. Figure 1 shows that the teacher–student likelihood ratio can identify better responses among student-generated candidates, and Appendix D shows the same for teacher likelihood. SMC allows us to use these signals to guide generation before a response is complete. Our algorithm, SMC-KD, uses teacher-guided SMC to construct training data without correctness labels or a separately trained reward model, and then transfers the resulting distribution into the student through supervised fine-tuning. We first define the sampling target and its potential, then describe the SFT projection and its connection to on-policy distillation.

3.1 SMC with teacher guidance

Let $\bar{\pi}_S = \pi_{\bar{\theta}}$ denote the student, held fixed during generation. We define a nonnegative reward on complete responses using teacher and student likelihoods, $r_{\alpha,\beta}(Y \mid q) = \bar{\pi}_S(Y \mid q)^\alpha \pi_T(Y \mid q)^\beta$. The choices $(\alpha, \beta) = (0, 1)$ and $(-1, 1)$ recover rewards based on teacher likelihood and the teacher–student likelihood ratio, respectively, and we refer to the latter as ratio guidance. Taking $\pi_{\mathrm{ref}} = \bar{\pi}_S$ and $r^* = r_{\alpha,\beta}$ in Equation 3 gives the target distribution

$$\pi^*_{\alpha,\beta}(Y \mid q) = \frac{\bar{\pi}_S(Y \mid q) r_{\alpha,\beta}(Y \mid q)}{Z^*_{\alpha,\beta}(q)} \propto \bar{\pi}_S(Y \mid q)^{1+\alpha} \pi_T(Y \mid q)^\beta, \tag{4}$$

where $Z^*_{\alpha,\beta}(q) = \mathbb{E}_{Y\sim\bar{\pi}_S(\cdot|q)}[r_{\alpha,\beta}(Y \mid q)]$ is assumed positive and finite. Because the student generates each response block by block, SMC needs a score for every prefix s_h. The ideal score is the continuation value described in Section 2.3, which averages the reward over the student's future blocks. For our reward, it splits into a prefix term and a continuation term,

$$V^*_{\alpha,\beta}(s_h) = \mathbb{E}_{y_{h+1:H}\sim\bar{\pi}_S(\cdot|s_h)}[r_{\alpha,\beta}(Y \mid q)] = \underbrace{\bar{\pi}_S(y_{1:h} \mid q)^\alpha \pi_T(y_{1:h} \mid q)^\beta}_{\widehat{V}_{\alpha,\beta}(s_h)} \underbrace{\sum_{z:\,\bar{\pi}_S(z|s_h)>0} \bar{\pi}_S(z \mid s_h)^{1+\alpha}\pi_T(z \mid s_h)^\beta}_{\kappa_{\alpha,\beta}(s_h)}. \tag{5}$$

Here z ranges over all possible continuations supported by the frozen student, and $\kappa_{\alpha,\beta}(s_h)$ is the resulting continuation factor. When the response is complete, no continuation remains, so $\kappa_{\alpha,\beta} = 1$ and $\widehat{V}_{\alpha,\beta} = r_{\alpha,\beta}$ on complete responses, as Section 2.3 requires. Computing $\kappa_{\alpha,\beta}(s_h)$ generally requires summing over all possible continuations, so we omit it and use $\widehat{V}_{\alpha,\beta}$ as an inexpensive approximation. Although $\kappa_{\alpha,\beta}$ varies across prefixes, omitting it therefore changes only the intermediate targets and not the final target. Additionally, for $(\alpha, \beta) = (-1, 1)$, the continuation factor reduces to the total teacher probability over student-supported continuations, which equals one when the student supports every teacher-supported response, making $\widehat{V}_{-1,1} = V^*_{-1,1}$ at every prefix.

We instantiate Algorithm 1 with proposal $\bar{\pi}_S$ and potential $\widehat{V}_{\alpha,\beta}$. Each resampled prefix $\widetilde{s}^i_{h-1}$ is extended by a student-generated block y^i_h, giving $s^i_h = [\widetilde{s}^i_{h-1}, y^i_h]$ and incremental weight $w^i_h = \widehat{V}_{\alpha,\beta}(s^i_h)/\widehat{V}_{\alpha,\beta}(\widetilde{s}^i_{h-1}) = \bar{\pi}_S(y^i_h \mid \widetilde{s}^i_{h-1})^\alpha \pi_T(y^i_h \mid \widetilde{s}^i_{h-1})^\beta$. Notably, the weight uses only the new block's likelihoods conditioned on that prefix, since the accumulated prefix likelihoods cancel. Normalizing these weights and resampling directs subsequent generation toward higher-weight particles. After the last round, the weighted population $\widehat{\nu}_H$ approximates the target in Equation 4.

3.2 Weighted SFT projection

We transfer the target distribution into the student's parameters by minimizing the forward KL divergence over the prompt distribution $\mathcal{Q}$, $\mathcal{L}_{\mathrm{KD}}(\theta;\bar{\theta}) = \mathbb{E}_{q\sim\mathcal{Q}}\Big[D_{\mathrm{KL}}\Big(\pi^*_{\alpha,\beta}(\cdot \mid q) \,\|\, \pi_\theta(\cdot \mid q)\Big)\Big]$, where θ are the student parameters being trained and $\bar{\theta}$ are the frozen parameters that define $\pi^*_{\alpha,\beta}$ through $\bar{\pi}_S$. Minimizing $\mathcal{L}_{\mathrm{KD}}$ amounts to maximizing the student likelihood of responses drawn from $\pi^*_{\alpha,\beta}$. For B prompts, let $\{Y^{b,i}, \omega^{b,i}\}_{i=1}^N$ denote the responses and normalized weights of the final population $\widehat{\nu}_H$ for prompt q^b. The particle estimate of this loss is

$$\widehat{\mathcal{L}}_{\mathrm{KD}}(\theta;\bar{\theta}) = -\frac{1}{B}\sum_{b=1}^{B}\sum_{i=1}^{N}\omega^{b,i}\log\pi_\theta(Y^{b,i} \mid q^b). \tag{6}$$

Because SMC only approximates $\pi^*_{\alpha,\beta}$ with a finite population, this estimate fits the distribution of SMC outputs, which approaches $\pi^*_{\alpha,\beta}$ as N grows and as $\widehat{V}_{\alpha,\beta}$ approaches $V^*_{\alpha,\beta}$. During training, the responses and weights are treated as fixed supervision, and since the teacher's influence is already encoded in them, the optimization requires only the student and ordinary next-token likelihood training.

Training alternates between constructing these populations and fitting the student to them. We collect responses from a fixed student snapshot, take U optimizer steps on their weighted likelihood, and then refresh the generating student, repeating this K times. The number of prompts B collected per refresh and the number of updates U are hyperparameters, allowing the frequency of data generation and model updates to be chosen independently of the sampling rule. Algorithm 2 summarizes this procedure. With $K = 1$ and B equal to the size of the prompt pool, SMC is run once over all prompts with the base student, and the student is trained on the result. We call this setting one-shot SMC-KD and refer to $K > 1$ as iterative SMC-KD. Because SMC-KD uses no correctness labels, the selected responses can still be poor, and with $K > 1$, training on them can degrade the responses generated in later iterations. We observe this with the Qwen2.5-0.5B student, and using $K = 1$ avoids it (Section 4.3 and Appendix E.2).

Algorithm 2 SMC-KD

Require: Student π_θ, teacher π_T, prompt pool $\mathcal{Q}$, reward exponents (α, β), particles N, block size C, maximum blocks H, prompts per refresh B, updates per refresh U, refresh iterations K

for $k = 1, \ldots, K$ **do**
  Set $\bar{\theta} \leftarrow \theta$ and freeze $\bar{\pi}_S = \pi_{\bar{\theta}}$
  Sample prompts $q^{1:B}$ from $\mathcal{Q}$
  for each prompt q^b **do**
    Run Algorithm 1 on q^b with $\bar{\pi}_S$, $\widehat{V}_{\alpha,\beta}$ (using π_T), N particles, and H blocks of C tokens
    Retain final responses and normalized weights $\{Y^{b,i}, \omega^{b,i}\}_{i=1}^N$
  end for
  Update θ with U optimizer steps on Equation 6
end for
return π_θ

3.3 Connection to on-policy distillation

The sampling and projection steps admit a policy-improvement interpretation that connects SMC-KD to OPD. For a fixed prompt, the target in Equation 4 solves $\pi^*_{\alpha,\beta} = \arg\max_\rho \{\mathbb{E}_{Y\sim\rho}[\log r_{\alpha,\beta}(Y \mid q)] - D_{\mathrm{KL}}(\rho \,\|\, \bar{\pi}_S)\}$. For ratio guidance $(\alpha, \beta) = (-1, 1)$, and assuming that the student supports every teacher-supported response, the target is $\pi^*_{-1,1} = \pi_T$, and the reference terms cancel in the improvement objective:

$$\mathbb{E}_{Y\sim\rho}\left[\log \frac{\pi_T(Y \mid q)}{\bar{\pi}_S(Y \mid q)}\right] - D_{\mathrm{KL}}(\rho \,\|\, \bar{\pi}_S) = -D_{\mathrm{KL}}(\rho \,\|\, \pi_T). \tag{7}$$

Thus ratio-guided SMC has the same distribution-level improvement objective as reverse-KL OPD (Equation 2). Our method, however, uses this signal at different points in learning. In sampled-token OPD updates, the teacher–student log-ratio supplies a token-level advantage (Yang et al., 2026). SMC-KD accumulates the log-ratio over each generated block and uses it to resample trajectories. Then weighted SFT fits the resulting population. Notably, the shared improvement objective does not imply identical parameter updates, because an exact SMC-KD projection minimizes the forward KL divergence $D_{\mathrm{KL}}(\pi_T \,\|\, \pi_\theta)$ (Equation 1) rather than the reverse KL optimized by OPD.

4 Experiments

In this section, we aim to answer two questions: 1) whether teacher-guided SMC generates better student responses than BoN, which Figure 1 briefly showed in the introduction and we examine here in more detail, and 2) whether distilling the resulting trajectories improves the student, and how it compares to on-policy distillation (OPD). Section 4.1 describes our experimental setup, Section 4.2 evaluates SMC as a generation procedure, and Section 4.3 evaluates the complete SMC-KD algorithm.

4.1 Experimental setup

Models. We use small students from four model families, Qwen2.5-0.5B (Qwen et al., 2025), Qwen3-0.6B (Yang et al., 2025), Gemma-3-1B (Team et al., 2025), and Llama-3.2-1B (Grattafiori et al., 2024), each paired with a larger teacher from the same family, namely Qwen2.5-Math-1.5B (Yang et al., 2024), Qwen3-1.7B, Gemma-3-4B, and Llama-3.2-3B, respectively. The distillation experiments use the Qwen2.5-0.5B and Llama-3.2-1B pairings, together with Qwen2.5-1.5B trained with Qwen2.5-Math-1.5B and Qwen2.5-0.5B trained with Qwen3-1.7B, a teacher from a different family.

Data. Following Lightman et al. (2024), we construct PRM12K from the MATH dataset (Hendrycks et al., 2021), combining all 7,500 original training problems with 4,500 original test problems. The remaining 500 test problems form MATH-500. We train only on the PRM12K problem statements, without reference solutions or correctness labels, and evaluate on MATH-500, the GSM8K test set (Cobbe et al., 2021), and AMC23 (Mathematical Association of America, 2023).

We then determine correctness with Hugging Face's Math-Verify evaluator (Kydlíček, 2025). Full hyperparameters are reported in Appendix B.

4.2 TEACHER-GUIDED SMC

In this section, we first compare SMC and BoN before any training to highlight the benefits of the SMC procedure. Both methods draw from the same frozen student and use the same population size N, however BoN generates N independent complete responses and selects the highest-scoring one, whereas SMC applies the score after every 64-token block to reweight and resample partial responses, and returns the final particle with the highest score, as BoN does. We evaluate two potentials: the student's own sequence log-likelihood, $\log \pi_S$, and the teacher–student log-likelihood ratio, $\log \pi_T - \log \pi_S$.

Table 1: **Accuracy (%) of BoN and SMC on MATH-500.** Within each cell, the top and bottom entries use self-likelihood and teacher–student likelihood-ratio guidance, respectively. pass@1 is accuracy without selection. BoN and SMC reports mean ± standard deviation over two seeds, and bold marks the higher mean between the two methods.

			$N = 4$		$N = 8$		$N = 16$		$N = 32$	
Samples from	Scored by	pass@1	BoN	SMC	BoN	SMC	BoN	SMC	BoN	SMC
Qwen2.5-0.5B	Self	28.2 ±1.4	32.8 ±0.0	**33.1** ±1.8	34.5 ±0.7	**35.2** ±0.6	36.1 ±1.0	**37.0** ±0.6	33.9 ±0.4	**37.1** ±0.7
	Qwen2.5-Math-1.5B		37.1 ±0.7	**40.0** ±1.4	40.0 ±1.1	**43.9** ±1.3	41.9 ±1.0	**47.7** ±0.4	45.1 ±0.1	**53.4** ±1.1
Qwen3-0.6B	Self	52.2 ±1.4	54.2 ±0.3	54.2 ±4.5	54.8 ±0.6	**56.4** ±1.4	54.1 ±0.4	**54.6** ±0.3	53.1 ±1.8	**53.5** ±0.4
	Qwen3-1.7B		59.5 ±1.6	**62.3** ±1.8	61.5 ±1.0	**65.3** ±0.4	64.2 ±1.4	**65.9** ±0.1	64.4 ±1.4	**67.7** ±0.4
Gemma-3-1B	Self	45.3 ±1.3	**50.0** ±3.1	45.4 ±1.1	**51.4** ±2.0	47.4 ±1.1	**50.4** ±0.8	47.6 ±0.8	**51.6** ±0.8	47.1 ±0.1
	Gemma-3-4B		51.7 ±3.3	**55.2** ±0.8	55.1 ±2.4	**58.6** ±2.3	56.6 ±1.7	**61.8** ±0.8	58.5 ±1.8	**60.8** ±1.7
Llama-3.2-1B	Self	21.3 ±1.5	**25.4** ±1.4	24.6 ±0.3	25.6 ±2.0	**26.0** ±0.3	26.2 ±0.8	**26.8** ±0.8	26.5 ±0.4	**27.8** ±1.7
	Llama-3.2-3B		29.8 ±0.3	**30.6** ±2.5	31.1 ±1.3	**33.1** ±2.4	33.7 ±1.0	**36.1** ±0.1	35.5 ±1.0	**38.2** ±0.0

Table 1 shows that teacher-guided SMC outperforms BoN for every model pairing and every population size. At $N = 8$, SMC improves accuracy by 3.9 percentage points for Qwen2.5-0.5B, 3.8 for Qwen3-0.6B, 3.5 for Gemma-3-1B, and 2.0 for Llama-3.2-1B. For Qwen2.5-0.5B, the gain grows to 8.3 points at $N = 32$ (53.4% versus 45.1%). These results show that applying teacher feedback to intermediate prefixes produces better responses than using the same score only after generation is complete.

Self-guided SMC, however, underperforms self-scored BoN for Gemma-3-1B at every population size and for Llama-3.2-1B at $N = 4$. Because SMC resamples on the potential after every block, it relies on the potential more than BoN does, and the student's own likelihood is not a strong enough signal to guide this resampling. Therefore, although self-guided SMC has been used for sampling from a sharpened sequence distribution (Azizi et al., 2026, Power-SMC), our results show that self-likelihood guidance can be brittle across models and particle counts. Appendix E.1 compares self, teacher-only, and likelihood-ratio potentials and varies the guidance strength across particle counts and student–teacher pairings.

4.3 SMC-KD

In this section, we evaluate the complete SMC-KD algorithm. Table 2 reports the accuracy of a single response *without* teacher access or inference-time search. All main SMC-KD results use the likelihood-ratio potential for every pairing. For Qwen2.5-1.5B and Llama-3.2-1B, we use iterative SMC-KD with $K = 12$ refresh iterations and $B = 1{,}024$ prompts per refresh, except for the last refresh, which uses 736 prompts so that the refreshes total 12,000 prompts, updating the student after each one. For both Qwen2.5-0.5B pairings, we use one-shot SMC-KD with $K = 1$ and $B = 12{,}000$, where SMC is run over the complete training set using the base student, followed by SFT.

In Table 2, we can see that SMC-KD matches or improves over the base student across the evaluated settings and outperforms OPD in 10 of the 12 student–teacher–benchmark comparisons, which shows that SMC-KD transfers the search improvement into more reliable one-shot responses across model families and evaluation sets. We highlight a key advantage of SMC-KD when the teacher

Table 2: **Accuracy (%) of the base student, OPD, and SMC-KD on three held-out math benchmarks.** SMC-KD uses the likelihood-ratio potential. Trained entries are mean ± standard deviation over three training seeds; bold marks the highest mean within each pairing and benchmark.

	MATH-500			GSM8K			AMC23		
Student ← Teacher	Base	OPD	SMC-KD	Base	OPD	SMC-KD	Base	OPD	SMC-KD
Qwen2.5-1.5B ← Qwen2.5-Math-1.5B	46.9 ±1.6	54.0 ±0.1	**55.9 ±0.3**	65.4 ±3.1	72.9 ±0.5	**73.9 ±1.1**	23.1 ±1.1	25.2 ±1.3	**26.0 ±1.0**
Llama-3.2-1B ← Llama-3.2-3B	21.4 ±1.3	24.2 ±0.4	**24.8 ±0.8**	37.1 ±1.6	38.8 ±1.1	**43.5 ±1.0**	10.1 ±0.4	11.4 ±0.5	**12.9 ±2.3**
Qwen2.5-0.5B ← Qwen2.5-Math-1.5B	28.1 ±1.4	32.0 ±0.2	**32.4 ±0.4**	40.8 ±1.7	**47.7 ±0.5**	46.5 ±0.6	10.2 ±0.5	**13.9 ±0.8**	12.7 ±0.4
Qwen2.5-0.5B ← Qwen3-1.7B	28.1 ±1.4	29.9 ±0.3	**31.1 ±0.1**	40.8 ±1.7	43.1 ±0.3	**46.0 ±0.1**	**10.2 ±0.5**	9.1 ±1.2	**10.2 ±1.5**

comes from a different model family. With the Qwen2.5-0.5B student, replacing the Qwen2.5-Math-1.5B teacher with Qwen3-1.7B reduces OPD accuracy on all three benchmarks, by 4.6 points on GSM8K and 4.8 points on AMC23, where OPD falls below the base student. Under the same change, SMC-KD loses at most 1.3 points on MATH-500 and GSM8K and remains above OPD on every benchmark. Liu et al. (2026) report similarly unstable reverse-KL distillation with a Qwen2.5 student and a Qwen3 teacher, which suggests that teacher–student distribution mismatch can degrade OPD. SMC-KD is likely less sensitive to this mismatch because it uses the teacher only to select among student-generated trajectories and then trains on them with SFT, rather than applying the teacher signal directly as a token-level update.

With the Qwen2.5-0.5B student, increasing the number of search-and-training rounds from $K = 1$ to $K = 12$ lowers accuracy on MATH-500 and AMC23 with both teachers, although it slightly improves GSM8K. Further investigation reveals repeated reasoning patterns in the iteratively trained model, together with a larger fraction of responses reaching the 2,048-token generation limit: on MATH-500, from 5.5% to 12.2% with Qwen2.5-Math-1.5B and from 5.3% to 16.0% with Qwen3-1.7B. The degradation is more pronounced with Qwen3-1.7B than with Qwen2.5-Math-1.5B. This pattern suggests that the stronger, math-specialized teacher can better distinguish promising trajectories from the increasingly degenerate responses produced by the iteratively trained student. When a weak student is paired with a teacher whose scores separate good and bad responses less reliably, repeated data collection can instead reinforce poor trajectories and compound the student's errors. Appendix E.2 reports both $K = 1$ and $K = 12$ for all four student–teacher pairings, including capped-response rates. It also compares the likelihood-ratio and teacher-only potentials on three pairings: the ratio gives the best student for the two same-family pairings, and the teacher-only potential for the cross-family pairing, by 0.2 points. This is consistent with Tables 4 and 5, where the ratio improves on the teacher's log-likelihood mainly when the student and teacher come from the same family.

5 Conclusion

In this work, we introduced SMC-KD, a knowledge distillation method that uses the teacher to select student-generated responses rather than to supervise individual tokens. SMC-KD guides SMC over the student's own block-by-block generation with a score based on teacher likelihood and trains the student on the resulting population through weighted SFT, without correctness labels or a separately trained reward model. Empirically, we showed that teacher-guided SMC finds more correct responses than BoN for every pairing and population size, and SMC-KD outperforms OPD in 10 of the 12 comparisons, including a cross-family pairing in which OPD degrades. Despite these promising results, our work also has limitations. First, as in SMC more generally, resampling reduces the diversity of the population by concentrating it on high-weight prefixes, and training on this population can also reduce the diversity of the student's responses. Second, because the potential multiplies likelihoods over all tokens of a response, its value depends on response length, which can bias selection toward shorter or longer responses regardless of their quality. Third, SMC-KD generates its training data with the student, so the data is limited to responses within the student's reach. Addressing these limitations, including preserving response diversity across iterations and accounting for response length in the potential, is left for future work.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods: large language model (LLM) coding assistants wrote parts of the code for our method and experiments. We have not used generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide ingredients for proofs or write proofs, propose or refine hypotheses, design research methodology or experiments, or interpret results; the research ideas, the method, the experimental design, and the claims of the paper are the authors' own. Generating synthetic datasets, translation, cleaning or reformatting datasets, and qualitative or thematic data analysis are not applicable to this work. The language models that our experiments sample from and distill are the objects of study, not tools used to conduct the research, so the traces they generate are not AI-assisted data in the sense of this statement.

Additionally, we used generative AI tools to create and edit software code, to draft parts of the paper, to edit the paper for readability, and to give feedback on the writing. We have reviewed all AI-assisted work. All AI-generated code was reviewed and tested for correctness by one author, and all AI-drafted or AI-edited text was checked and verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The sampling and training procedures are specified in Section 3, with plain SMC in Algorithm 1 and the full SMC-KD procedure in Algorithm 2; the resampling scheme and effective-sample-size rule are given in Appendix C. Section 4.1 lists the student and teacher models, the construction of the PRM12K training prompts from MATH, the evaluation benchmarks (MATH-500, GSM8K, and AMC23), and the answer-checking procedure; all models and datasets are publicly available. Appendix B reports the hyperparameters for all reported comparisons. We do not release code with this submission, but we plan to release it upon publication.

REFERENCES

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

Mohammad Sadegh Akhondzadeh, Vijay Lingam, Atula Tejaswi, Chanakya Ekbote, Sujay Sanghavi, and Aleksandar Bojchevski. Reward-gated on-policy distillation. *arXiv preprint arXiv:2607.04037*, 2026.

Seyedarmin Azizi, Erfan Baghaei Potraghloo, Minoo Ahmadi, Souvik Kundu, and Massoud Pedram. Power-smc: Low-latency sequence-level power sampling for training-free llm reasoning. *arXiv preprint arXiv:2602.10273*, 2026.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 1171–1179, Cambridge, MA, USA, 2015. MIT Press.

Nicolas Boizard, Kevin El Haddad, CELINE HUDELOT, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for LLMs. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=bwRxXiGO9A.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Robin Shing Moon Chan, Tianyu Liu, Samuel Kiegeland, Clemente Pasti, Jacob Hoover Vigly, Timothy J O'Donnell, Ryan Cotterell, and Tim Vieira. Ensembling language models with sequential monte carlo. *arXiv preprint arXiv:2603.05432*, 2026.

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Xiao Cui, Mo Zhu, Yulei Qin, Liang Xie, Wengang Zhou, and Houqiang Li. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 23724–23732, 2025.

Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pp. 64–69. Ieee, 2005.

Shengyu Feng, Xiang Kong, Shuang Ma, Aonan Zhang, Dong Yin, Chong Wang, Ruoming Pang, and Yiming Yang. Step-by-step reasoning for math problems via twisted sequential monte carlo. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ze4aPP0tIn.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

Giorgio Giannone, Mustafa Eyceoz, Shabana Baig, Shivchander Sudalairaj, Anna C Doris, Faez Ahmed, Akash Srivastava, and Kai Xu. Intrinsic selection and particle resampling for inference-time scaling beyond domain verifiability. *arXiv preprint arXiv:2606.08850*, 2026.

Noah Golowich, Fan Chen, Dhruv Rohatgi, Raghav Singhal, Carles Domingo-Enrich, Dylan J Foster, and Akshay Krishnamurthy. Reject, resample, repeat: Understanding parallel reasoning in language model inference. *arXiv preprint arXiv:2603.07887*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rStar-math: Small LLMs can master math reasoning with self-evolved deep thinking. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20640–20661. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/guan25f.html.

Lin Gui, Cristina Gârbacea, and Victor Veitch. Bonbon alignment for large language models and the sweetness of best-of-n sampling. *Advances in Neural Information Processing Systems*, 37: 2851–2885, 2024.

Chaoqun He, Yingfa Chen, Chaojun Xiao, Xu Han, and Lijie Wen. Student-in-the-loop chain-of-thought distillation via generation-time selection. *arXiv preprint arXiv:2604.02819*, 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 14852–14882, 2023.

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, et al. Uni-opd: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the association for computational linguistics: ACL 2023*, pp. 8003–8017, 2023.

Xiaotong Ji, Rasul Tutunov, Matthieu Zimmer, and Haitham Bou Ammar. Scalable power sampling: Unlocking efficient, training-free reasoning for LLMs via distribution sharpening. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=SVyjXhZlDe.

Yuxuan Jiang and Francis Ferraro. Bridging reasoning trajectories in on-policy distillation via near-future guidance. *arXiv preprint arXiv:2606.00305*, 2026.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=J5i09faOOf.

Hoang Anh Just, Myeongseob Ko, and Ruoxi Jia. The signal is in the steps: Local scoring for reasoning data selection, 2026. URL https://arxiv.org/abs/2510.03988.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. *Advances in neural information processing systems*, 38:19720–19745, 2026.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. In *International Conference on Learning Representations*, volume 2026, pp. 476–494, 2026.

Sooyeon Kim, Giung Nam, Byoungwoo Park, and Juho Lee. Improving constrained language generation via self-distilled twisted sequential monte carlo. *arXiv preprint arXiv:2507.02315*, 2025.

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

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31044–31062. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026.

Hynek Kydlíček. Math-Verify: Math verification library. Software, 2025. URL https://github.com/huggingface/Math-Verify.

Anh Duc Le, Tu Vu, Nam Le Hai, Nguyen Thi Ngoc Diep, Linh Ngo Van, Trung Le, and Thien Huu Nguyen. Cot2align: cross-chain of thought distillation via optimal transport alignment for language models with different tokenizers. *arXiv preprint arXiv:2502.16806*, 2025.

Alexander K. Lew, Tan Zhi-Xuan, Gabriel Grand, and Vikash Mansinghka. Sequential monte carlo steering of large language models using probabilistic programs. In *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*, 2023. URL https://openreview.net/forum?id=Ul2K0qXxXy.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026. URL https://openreview.net/forum?id=aaDDXXDGXB.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25366–25394, 2025.

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

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6121–6133, 2020.

Xinyu Liu, Kechen Jiao, Chunyang Xiao, Runsong Zhao, Junhao Ruan, Bei Li, Jiahao Liu, Qifan Wang, Xin Chen, Jingang Wang, et al. Teacher-guided policy optimization for on-policy reasoning distillation under large policy divergence. *arXiv preprint arXiv:2605.13230*, 2026.

João Loula, Benjamin LeBrun, Li Du, Ben Lipkin, Clemente Pasti, Gabriel Grand, Tianyu Liu, Yahya Emara, Marjorie Freedman, Jason Eisner, et al. Syntactic and semantic control of large language models via sequential monte carlo. In *International Conference on Learning Representations*, volume 2025, pp. 64758–64791, 2025.

Jelena Markovic-Voronov, Wenhui Zhu, Bo Long, Zhipeng Wang, Suyash Gupta, Kayhan Behdin, Bee-Chung Chen, and Deepak Agarwal. Sampling for quality: Training-free reward-guided llm decoding via sequential monte carlo. *arXiv preprint arXiv:2604.16453*, 2026.

Mathematical Association of America. 2023 american mathematics competitions, 2023. URL https://maa.org/student-programs/amc/.

Benjamin Minixhofer, Ivan Vulić, and Edoardo Maria Ponti. Universal cross-tokenizer distillation via approximate likelihood matching. *Advances in Neural Information Processing Systems*, 38: 79297–79326, 2026.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. Controlled decoding from language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Jingyu Peng, Maolin Wang, Hengyi Cai, Yuchen Li, Kai Zhang, Shuaiqiang Wang, Dawei Yin, and Xiangyu Zhao. Adaswitch: Balancing exploration and guidance in knowledge distillation via adaptive switching, 2026. URL https://arxiv.org/abs/2510.07842.

Isha Puri, Shivchander Sudalairaj, Guangxuan Xu, Abhishek Bhandwaldar, Kai Xu, and Akash Srivastava. Rollout roulette: A probabilistic inference approach to inference-time scaling of llms using particle-based monte carlo methods. *Advances in Neural Information Processing Systems*, 38:156427–156454, 2026.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence Level Training with Recurrent Neural Networks. In *International Conference on Learning Representations*, 2016. URL https://mlanthology.org/iclr/2016/ranzato2016iclr-sequence/.

Pier Giuseppe Sessa, Robert Dadashi-Tazehozi, Leonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Rame, Bobak Shahriari, Sarah Perrin, Abram L. Friesen, Geoffrey Cideron, Sertan Girgin, Piotr Stanczyk, Andrea Michi, Danila Sinopalnikov, Sabela Ramos Garea, Amélie Héliou, Aliaksei Severyn, Matthew Hoffman, Nikola Momchev, and Olivier Bachem. BOND: Aligning LLMs with best-of-n distillation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0tAXMiSufG.

Avi Singh, John D Co-Reyes, and Rishabh Agarwal. Beyond human data: Scaling self-training for problem-solving with language models. In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*, 2024. URL https://openreview.net/forum?id=5pGUyc4PeW.

Avyav Kumar Singh, Yen-Chen Wu, Alexandru Cioba, Alberto Bernacchia, and Davide Buffelli. Cross-tokenizer llm distillation through a byte-level interface. In *Proceedings of the Second Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pp. 84–96, 2026.

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

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Bing Wang, Shaotian Yan, Chen Shen, Sinan Fan, Ximing Li, Rui Miao, Xiaosong Yuan, Zhanming Shen, Jieping Ye, et al. Backtracking when it strays: Mitigating dual exposure biases in llm reasoning distillation. *arXiv preprint arXiv:2605.19433*, 2026.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yecheng Wu, Song Han, and Hai Cai. Lightning opd: Efficient post-training for large reasoning models with offline on-policy distillation. *arXiv preprint arXiv:2604.13010*, 2026.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, volume 2025, pp. 64616–64646, 2025.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*, 2024.

JianZhi Yan, Le Liu, Youcheng Pan, Shiwei Chen, Yang Xiang, and Buzhou Tang. Towards efficient CoT distillation: Self-guided rationale selector for better performance with fewer rationales. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 7818–7835, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.413. URL https://aclanthology.org/2025.findings-emnlp.413/.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

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

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Ping Yu, Jing Xu, Jason E Weston, and Ilia Kulikov. Distilling system 2 into system 1. In *The First Workshop on System-2 Reasoning at Scale, NeurIPS'24*, 2024. URL https://openreview.net/forum?id=WUoC4BpJBC.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772, 2024.

Dylan Zhang, Qirun Dai, and Hao Peng. The best instruction-tuning data are those that fit. *Advances in Neural Information Processing Systems*, 38:141172–141208, 2026a.

Ke Zhang, Yunjie Tian, Dongdi Zhao, Yijiang Li, Yuanye Liu, Vishal M Patel, and Di Fu. On-policy distillation with best-of-n teacher rollout selection. *arXiv preprint arXiv:2605.09725*, 2026b.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516, 2025.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. probabilistic inference in language models via twisted sequential monte carlo. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Zhou Ziheng, Jiaqi Li, Huacong Tang, Ying Nian Wu, and Demetri Terzopoulos. Less is more: Early stopping rollout for on-policy distillation. *arXiv preprint arXiv:2605.27028*, 2026.

A Extended Related Work

We organize prior work along four dimensions of our pipeline. First, we consider who generates the training responses. Second, we examine how the teacher provides supervision over contexts generated by the student. Third, we distinguish methods that perform selection only over complete responses from those that also operate on partial prefixes. Fourth, we consider whether the guidance is provided by an external teacher or derived from the student itself. These dimensions provide a common framework for relating sequence-level knowledge distillation, on-policy distillation (OPD), guided sequential Monte Carlo (SMC), and self-guided sampling. We further compare these approaches in terms of their target distributions, teacher-access requirements, and the stage at which computational cost is incurred.

A.1 Knowledge distillation and student learnability

Knowledge distillation transfers teacher behavior to a student through soft targets or generated text. Classical KD matched the teacher's softened output distribution (Hinton et al., 2015). Sequence-level KD instead trained on complete teacher-decoded sequences (Kim & Rush, 2016). Reasoning distillation used teacher-written rationales and explanations (Ho et al., 2023; Hsieh et al., 2023). These methods provide complete training targets, but the contexts come from the teacher. At inference, the student conditions on its own predictions. An early error can move it to a state absent from the training data.

Data selection addresses a second problem: a strong teacher's response may be difficult for a small student to learn. GRAPE collects candidate responses from several sources. It selects the response with the highest length-normalized likelihood under the target student (Zhang et al., 2026a). Just et al. (2026) found that whole-response scoring failed on long reasoning traces from heterogeneous teachers. They instead scored local reasoning transitions using a short context window. MoRSD selects rationales based on correctness, diversity, and student difficulty (Yan et al., 2025). Its difficulty score measures how much a rationale helps the student predict the correct answer. Li et al. (2025) reported a learnability gap in models with roughly 3B parameters or fewer. They addressed it by mixing short chains with long teacher-written chains.

Gen-SSD moves student-aware selection into the teacher's generation process (He et al., 2026). The teacher generates partial continuations, and the student selects among them using its own perplexity. SMC-KD reverses these roles. The student generates every token, while teacher scores guide prefix resampling. This produces teacher-preferred examples from the student's own distribution. We compare this prefix-level selection with BoN, which selects among the same student's responses only after they are complete.

Beyond choosing examples that fit the student, distillation can train a model to reproduce the results of sample selection. BOND iteratively trained a policy to match the distribution produced by reward-based Best-of-N (BoN) selection (Sessa et al., 2025). BoNBoN learned from preferences between ranked samples to approximate the BoN distribution (Gui et al., 2024). Self-distilled twisted SMC used samples from one round to update the proposal and twist for the next round (Kim et al., 2025). It improved constrained generation while retaining particle sampling at inference.

SMC-KD also uses guided sampling to construct student-generated training data. A fixed teacher scores student prefixes during data generation. Supervised fine-tuning (SFT) then trains on the resulting responses without further teacher queries. In the iterative setting, the updated student generates a new batch in the next round, while the teacher remains fixed. OPD instead uses teacher feedback to supervise the student's predictions during the distillation update.

A.2 On-policy distillation and teacher feedback

OPD uses teacher feedback to supervise the student on contexts it generates rather than only on teacher-written sequences. ImitKD obtained teacher corrections on model-generated contexts (Lin et al., 2020), whereas GKD optimized divergences between teacher and student on student rollouts (Agarwal et al., 2024). MiniLLM minimized reverse KL from the student to the teacher on student samples (Gu et al., 2024). This discourages the student from assigning probability to outputs that the teacher considers unlikely, but it can concentrate the student's distribution on a few modes (Jin et al.,

2026). DistiLLM changed the divergence and reused samples (Ko et al., 2024), while DistiLLM-2 combined teacher- and student-generated data (Ko et al., 2025). On-policy context distillation used a context-conditioned teacher on student rollouts (Ye et al., 2026). These methods expose student-written contexts that teacher-only data may miss, but they still need useful teacher feedback on those contexts.

That feedback may weaken as student-written prefixes drift from teacher-generated text: Fu et al. (2026) found uneven token-level supervision and unreliable guidance on such prefixes. Early Stopping Rollout likewise found weaker guidance later in a rollout and limited OPD to early tokens (Ziheng et al., 2026). A large token loss does not always mark a wrong reasoning step: trajectory-aware OPD found that different wording can receive high loss even when the reasoning path is similar. It used near-future continuations to distinguish such cases from genuine drift (Jiang & Ferraro, 2026). Li et al. (2026) linked successful transfer to reasoning patterns shared by teacher and student, while Hou et al. (2026) checked whether teacher token preferences agree with outcome rewards.

When teacher and student disagree during generation, the teacher can also write part of the continuation. Speculative KD can replace rejected student proposals with teacher tokens (Xu et al., 2025), while AdaSwitch chooses between student and teacher generation based on their divergence (Peng et al., 2026). MOTAB backtracks from a drifting prefix before adding a teacher continuation (Wang et al., 2026); teacher-guided policy optimization requests one when the policy gap is large (Liu et al., 2026). These methods can produce traces with teacher-written text, and their interventions act on one trajectory at a time. SMC-KD keeps the student as the sole generator. The teacher instead scores a population of student-written prefixes and guides which prefixes receive more generation.

Instead of writing tokens directly, teacher feedback can also shape the training loss or select the training examples. Teacher top-K local support matching compares teacher-supported tokens rather than the full vocabulary (Fu et al., 2026). Reverse KL can narrow the student's output distribution when the teacher is uncertain, so entropy-aware OPD adds forward KL at high-entropy teacher positions to retain several plausible next tokens (Jin et al., 2026). BRTS selects teacher rollouts by correctness and student alignment before adding an auxiliary OPD branch (Zhang et al., 2026b). Reward-gated OPD uses verifier outcomes to decide when to trust dense teacher logits (Akhondzadeh et al., 2026). SMC-KD uses teacher scores to choose student-written responses for SFT, and its SFT loss uses no teacher logits.

But full teacher logits are not always available for these losses. Black-box OPD learns a discriminator from teacher-generated text (Ye et al., 2025), while Lightning OPD stores the teacher probability of the student-selected token under an offline-consistency condition (Wu et al., 2026). Dense token-level matching is harder when teacher and student use different tokenizers, or handle special tokens differently (Fu et al., 2026). ULD and MultiLevelOT align token distributions with optimal transport (Boizard et al., 2025; Cui et al., 2025). CoT2Align aligns reasoning chains (Le et al., 2025); approximate likelihood matching compares aligned text blocks (Minixhofer et al., 2026); byte-level distillation uses a shared byte interface (Singh et al., 2026). SMC-KD needs scores for student-written prefixes, not aligned token distributions or teacher-written continuations. Each model tokenizes the same prefix text independently. Teacher scoring during data generation remains necessary.

When the teacher can score student-written prefixes, the score can guide a policy update or the construction of training data. Generalized OPD uses the teacher-to-reference log-probability ratio as a reward for policy updates (Yang et al., 2026). Relaxed OPD clips that reward and updates selected tokens (Ko et al., 2026). SMC-KD instead uses teacher scores during generation to resample student-written prefixes. This changes which responses receive more generation before SFT. Once the data have been constructed, SFT proceeds without further teacher access.

A.3 Sequential Monte Carlo for guided generation

Choosing which partial response to continue requires a score before generation ends. Outcome verifiers rank completed solutions and therefore cannot guide that choice directly (Cobbe et al., 2021). Process supervision supplies intermediate scores through human labels, model-generated labels, or rollout estimates (Uesato et al., 2022; Lightman et al., 2024; Wang et al., 2024). Labels inferred from final outcomes can be misleading, however, a flawed step may receive credit because the completed answer is correct (Zhang et al., 2025). Controlled decoding takes another route by learning a prefix value for sampling from a reward-tilted distribution (Mudgal et al., 2024). These

approaches depend on correctness labels, a task reward, or a trained value model. SMC-KD instead scores student-written prefixes with teacher likelihood, without requiring reasoning-step boundaries.

Given a score for partial responses, SMC can shift generation toward promising prefixes. It maintains a population of prefixes, updates their weights, and resamples them as generation proceeds. SMC steering applied this mechanism to general potentials in language-model programs (Lew et al., 2023). Later work used it for syntactic and semantic constraints (Loula et al., 2025). SMC ensembles use sequence-level targets to combine language models, with a shared byte space allowing different vocabularies (Chan et al., 2026). In each case, resampling acts on the evolving sequence rather than an average of next-token predictions.

The source of the prefix score matters because many objectives are defined only for complete responses. Twisted SMC learns a twist that estimates the expected future value of a terminal potential, allowing that potential to guide resampling before the sequence ends (Zhao et al., 2024). A reasoning variant estimates future rewards from outcome-supervised solutions (Feng et al., 2025). When intermediate rewards are available, Rollout Roulette uses process reward scores to weight and resample reasoning prefixes (Puri et al., 2026), reward-guided SMC uses prefix-dependent reward potentials (Markovic-Voronov et al., 2026). SMC-KD scores each student-written block with teacher likelihood, so it does not train a twist or process reward model.

These prefix-level scores determine how the particle population evolves, but a better particle approximation does not imply better answers by itself. Golowich et al. (2026) gave non-asymptotic guarantees and limits for myopic particle methods. They also showed that small sampling error does not ensure high answer accuracy. In SMC-KD, the particle population approximates the teacher-guided target, and the responses used for SFT are obtained from this population. The theoretical guarantee only shows that this sampled distribution can be close to the target distribution. It does not guarantee that the sampled responses are more accurate, or that training on them will necessarily improve the student model.

This separation between distributional approximation and answer quality also motivates a comparison with terminal-only selection. BoN returns the completed response with the highest scalar score. Its selected distribution has a reward–KL trade-off (Beirami et al., 2025). Our teacher-scored BoN baseline uses the same student, teacher score, particle count, and final selection rule as SMC. But only SMC resamples intermediate prefixes. The direct-generation comparison therefore isolates when the teacher signal is applied: Best-of-N uses it only after complete responses have been generated, whereas SMC uses it throughout generation. Importantly, both methods still rely on an external scoring model. The fact that the student generates the responses does not make either method self-guided.

A.4 Self-guided sampling and self-training

Self-guided generation selects responses using signals from the model or agreement among its outputs. Repeated sampling can find solutions missed by a single attempt. Oracle $\mathrm{pass@}\,k$ counts a problem as solved if any of k samples is correct, and it continued to improve even after practical selection methods plateaued (Brown et al., 2024). Because it assumes that the correct sample can be identified, it measures sample coverage rather than providing a practical selection rule. Self-consistency provides such a rule by sampling several reasoning paths and returning the most common final answer (Wang et al., 2023). This requires answers that can be normalized and grouped. Self-certainty avoids this requirement by scoring each response from the model's next-token distributions, with more concentrated distributions receiving higher scores (Kang et al., 2026). Neither method uses an external teacher. Beyond selecting completed responses, extra compute can also be used to generate candidates in different ways. Snell et al. (2024) compared parallel sampling, sequential answer revision, and verifier-guided search under a fixed budget, and found that the best choice depended on problem difficulty. Their work separates two decisions: how to generate candidates and how to select among them.

The previous methods change how completed responses are selected or how the compute budget is spent. Another option is to change the distribution from which responses are sampled. Analyses of reasoning RL found that many gains came from increasing the probability of solution patterns already present in the base model (Chen et al., 2026). Power sampling applies this idea at inference by assigning more probability to sequences that the model already considers likely. Karan

& Du (2026) used Markov chain Monte Carlo to sample from this sharpened distribution. Later work approximated the same target with autoregressive decoding (Ji et al., 2026) or SMC (Azizi et al., 2026). Intrinsic particle filtering instead used the generator's own entropy to rank completed responses and resample partial responses (Giannone et al., 2026). These methods rely on signals from the generator itself. SMC-KD also resamples partial responses, but its weights come from an external teacher.

The methods above use sampling and selection to improve a response. Self-training uses the selected responses to update the model, but the selection step may still require external supervision. STaR kept rationales that led to correct answers and regenerated failed rationales after providing the known answer (Zelikman et al., 2022). Rejection-sampling fine-tuning retained generated solutions that passed answer or program checks (Yuan et al., 2023; Singh et al., 2024). ReST-MCTS and rStar-Math used tree search with outcome or process signals to construct training traces (Zhang et al., 2024; Guan et al., 2025). Yu et al. (2024) instead filtered outputs using agreement or invariance under input changes, without external labels. These methods all train on model-generated text, but they use different signals to decide which examples to keep.

SMC-KD also trains on student-generated responses, but the guidance used to construct this data comes from an external teacher. Within each data-generation round, the student and teacher remain frozen: the student generates particles, and the teacher scores their prefixes. The resulting student-written responses provide supervision for SFT, which uses no further teacher queries. In the iterative setting, the updated student then generates the next batch, while the teacher remains fixed. At deployment, standard decoding uses neither a teacher nor particles. This distinguishes SMC-KD from the intrinsic methods above, which rely only on signals from the generator itself.

B HYPERPARAMETERS

Table 3 lists the settings used for the reported comparisons.

Table 3: Hyperparameters

Setting	Value
Sampling (BoN, direct-generation SMC, evaluation)	Hugging Face defaults of each model
Maximum response length	2,048 tokens
SMC block size	64 tokens
Guidance strength	$\lambda = 1$
Resampling	Systematic, when ESS $< N/2$
SMC-KD particles	$N = 8$
SFT batch	32 prompts, all weighted particles
SFT learning rate	5×10^{-6}
OPD loss	Full-vocabulary reverse KL
OPD batch	16 prompts $\times$ 8 rollouts
OPD learning rate	2×10^{-6}
Optimizer updates (SMC-KD and OPD)	750
Schedule	Constant, 5% warmup, AdamW, weight decay 0.01
Evaluation samples per seed	8 (MATH-500, AMC23), 1 (GSM8K)

C SYSTEMATIC RESAMPLING

Let $\bar{w}^i = w^i / \sum_{j=1}^N w^j$ be the normalized particle weights and let $F_i = \sum_{j=1}^i \bar{w}^j$ be their cumulative sum. Systematic resampling draws one offset $u \sim \mathrm{Uniform}[0, 1/N)$ and forms the evenly spaced points $u_k = u + (k-1)/N$ for $k = 1, \ldots, N$. The ancestor of particle k is the smallest index i such that $F_i \geq u_k$. This produces exactly N offspring while using one random offset instead of N independent categorical draws, typically reducing resampling noise in the particle population. The effective sample size is $\mathrm{ESS} = (\sum_i w^i)^2 / \sum_i (w^i)^2$, which equals N for uniform weights and 1 when a single particle holds all the weight. When ESS $\geq N/2$, we skip resampling and each particle keeps its weight, multiplied by the next incremental weight.

D WHOLE-RESPONSE LOG-LIKELIHOOD SCORING

Tables 4 and 5 give the complete generator–scorer results summarized in Figure 1. They address a question that precedes guided search: given complete responses from a generator, which model's likelihood best identifies the correct ones? We sample 32 responses per MATH-500 prompt from each of thirteen models in four families: Qwen2.5 (0.5B, 1.5B, 3B, and 7B, together with Qwen2.5-Math-1.5B), Qwen3 (0.6B, 1.7B, 4B, and 8B), Gemma 3 (1B and 4B), and Llama 3.2 (1B and 3B). Every model then scores every response, reading the response text with its own tokenizer and chat template, so the likelihood is defined for every pair, including pairs whose tokenizers differ. For each generator–scorer pair, we return the response with the highest scorer log-likelihood $\log \pi_T(Y \mid q)$ or with the highest log-likelihood ratio $\log \pi_T(Y \mid q) - \log \pi_S(Y \mid q)$, which is the terminal score of the ratio potential in SMC-KD. When the scorer is the generator itself, the first criterion is self-scored BoN and the ratio is identically zero. Table 4 reports BoN@8, and Table 5 reports BoN@32. Selection improves on pass@1 for nearly every pair, but the size of the gain depends strongly on the scorer.

The most visible pattern in both tables is a block structure along the diagonal: the best scorer for a generator is almost always a model from its own family, typically a larger one, rather than the strongest model overall. For the nine generators with a larger model in their own family, the best scorer comes from that family in 35 of the 36 combinations of criterion and budget; the exception is Llama-3.2-1B under $\log \pi_T$ at $N = 32$, where Qwen2.5-7B exceeds Llama-3.2-3B by 0.8 points. Solving ability does not transfer to scoring responses from another family. Qwen3-8B has the highest pass@1 of the thirteen models (84.3%), yet Qwen2.5-1.5B, at 46.9%, is a better scorer for Qwen2.5-0.5B under both criteria and at both budgets. Likewise, Llama-3.2-3B, at 41.0%, is a better scorer for Llama-3.2-1B, reaching 34.8% at $N = 32$ under the ratio, compared with 26.8% for Qwen3-8B. A scorer from another family can even make selection worse than a single sample: every entry below pass@1 (red) pairs a generator with a scorer from a different family. A plausible explanation is that models from the same family share a tokenizer and much of their training data and response style, so a same-family scorer's likelihood reflects what a response says more than how it is written. The two generators that do not follow the pattern, Llama-3.2-3B and Gemma-3-4B, have no larger model in their family. Qwen2.5-7B is the best scorer for Llama-3.2-3B, and the accuracy of Gemma-3-4B varies by at most 2 points across all thirteen scorers.

Within a family, the size of the scorer also matters: selection accuracy generally increases with scorer size, most clearly for the weakest generators, which have the most room for improvement. At $N = 32$ under $\log \pi_T$, the accuracy of Qwen2.5-0.5B rises from 34.2% to 39.0%, 41.8%, and 43.0% as the Qwen2.5 scorer grows from 0.5B to 1.5B, 3B, and 7B, and the accuracy of Qwen3-0.6B rises from 54.4% to 63.2%, 63.6%, and 64.2% as the Qwen3 scorer grows from 0.6B to 8B. Gemma-3-1B and Llama-3.2-1B are likewise scored better by their larger sibling than by themselves at both budgets.

Family membership also determines which criterion works better: the log-likelihood ratio improves on the scorer's log-likelihood mainly when the two models come from the same family. Among the 36 same-family pairs off the diagonal, the ratio selects better in 24 at each budget and raises accuracy by 0.6 points on average at $N = 8$ and by 1.1 points at $N = 32$. The largest gains are for the weakest generators: at $N = 32$, the best ratio entry exceeds the best $\log \pi_T$ entry by 5.2 points for Qwen3-0.6B, 3.8 points for Llama-3.2-1B, and 3.4 points for Qwen2.5-0.5B. Across families, the ratio selects worse in 89 of the 120 pairs at each budget and lowers accuracy by 0.3 points on average at $N = 8$ and by 0.4 points at $N = 32$. We return to this difference for the cross-family SMC-KD pairing in Appendix E.2.

The pairings in our main experiments are consistent with these results. Each student is paired with a teacher from its own family, and for the Qwen2.5 students the teacher is Qwen2.5-Math-1.5B, which scores Qwen2.5 generators about as well as or better than Qwen2.5-3B, a general-purpose model twice its size. The one exception, Qwen2.5-0.5B $\leftarrow$ Qwen3-1.7B, is included deliberately to show what happens when the teacher is changed from one family to another while the student stays the same. According to Table 4, this cross-family teacher is a weaker scorer for Qwen2.5-0.5B than Qwen2.5-Math-1.5B, with a BoN@8 of 35.0% versus 39.9% under the ratio.

Table 4: BoN@8 accuracy (%) on MATH-500 for every generator–scorer pair. In each cell, the top entry selects by the scorer's log-likelihood $\log \pi_T$ and the bottom entry by the ratio $\log \pi_T - \log \pi_S$, where π_S is the generator; entries are mean $\pm$ standard deviation over four disjoint groups of 8 samples. M-1.5B is Qwen2.5-Math-1.5B. Bold marks the best scorer for each generator and criterion, and outlines mark same-family blocks. Shading shows the share of the gap between pass@1 and pass@8 that selection closes; red marks selection below pass@1.

			Scored by Qwen2.5					Scored by Qwen3				Scored by Gemma 3		Scored by Llama 3.2	
Samples from	pass@1	pass@8	0.5B	1.5B	3B	7B	M-1.5B	0.6B	1.7B	4B	8B	1B	4B	1B	3B
Qwen2.5-0.5B	28.1 ±0.2	57.5 ±0.7	34.6 ±0.6 –	36.5 ±1.8 37.7 ±1.8	38.7 ±1.2 39.9 ±0.9	**39.8 ±1.7** **41.2 ±1.2**	38.3 ±1.4 39.9 ±0.9	33.7 ±0.7 33.1 ±0.9	35.3 ±0.6 35.0 ±1.5	36.0 ±0.8 36.2 ±1.0	35.4 ±0.6 35.2 ±1.3	32.9 ±0.8 32.6 ±1.5	36.3 ±0.7 36.5 ±0.8	33.5 ±0.4 33.1 ±0.5	35.9 ±0.7 35.8 ±1.1
Qwen2.5-1.5B	46.9 ±0.7	75.0 ±1.5	52.8 ±2.8 51.8 ±1.9	52.8 ±2.8 –	55.7 ±2.4 57.0 ±1.6	**56.6 ±2.6** **58.8 ±2.4**	56.0 ±2.4 57.4 ±2.1	50.6 ±3.4 50.0 ±3.1	52.1 ±3.5 52.0 ±3.5	53.0 ±2.9 52.9 ±3.9	53.0 ±3.4 52.4 ±3.1	52.4 ±2.7 51.8 ±2.0	54.1 ±3.2 53.5 ±3.1	51.6 ±1.8 49.1 ±2.1	52.6 ±2.8 51.4 ±2.7
Qwen2.5-3B	65.5 ±0.7	85.2 ±0.4	68.5 ±0.8 68.0 ±0.7	69.4 ±0.7 68.9 ±1.3	69.0 ±1.1 –	**71.2 ±0.7** **71.5 ±1.3**	70.5 ±0.9 **71.5 ±1.6**	67.7 ±0.9 66.8 ±1.0	68.6 ±0.9 68.1 ±0.7	69.0 ±0.6 68.5 ±0.9	69.5 ±1.1 69.0 ±1.2	68.3 ±1.5 68.0 ±1.9	69.5 ±1.1 69.1 ±1.2	68.6 ±0.7 68.2 ±0.3	69.5 ±0.8 68.9 ±0.8
Qwen2.5-7B	75.8 ±0.3	89.7 ±0.9	77.7 ±0.9 77.0 ±0.9	78.4 ±0.8 77.2 ±0.6	78.5 ±0.8 78.4 ±0.4	78.7 ±1.1 –	**78.8 ±0.7** **79.1 ±1.0**	77.8 ±0.9 77.3 ±0.8	77.9 ±0.6 77.6 ±0.6	78.5 ±1.0 78.6 ±0.8	78.6 ±0.5 78.4 ±0.8	77.0 ±0.8 77.4 ±0.7	78.0 ±0.9 77.5 ±1.0	76.7 ±1.0 76.5 ±0.4	77.1 ±0.8 77.0 ±0.7
Q2.5-Math-1.5B	75.0 ±0.3	88.6 ±0.7	76.0 ±1.1 76.1 ±0.8	76.9 ±1.2 76.8 ±1.4	77.7 ±0.9 78.0 ±0.9	**78.8 ±1.1** **79.3 ±1.2**	77.5 ±1.0 –	75.9 ±1.5 75.8 ±1.3	76.2 ±0.8 76.2 ±0.8	76.2 ±0.5 76.4 ±0.6	76.5 ±0.9 76.6 ±1.1	75.8 ±1.1 75.3 ±1.0	77.0 ±1.1 77.1 ±1.2	76.2 ±0.9 75.9 ±0.5	77.1 ±0.8 76.6 ±1.0
Qwen3-0.6B	52.3 ±0.4	79.9 ±0.8	52.8 ±1.4 51.7 ±1.6	53.6 ±1.3 52.6 ±1.8	55.0 ±1.1 54.5 ±2.0	55.7 ±1.7 56.4 ±2.0	54.8 ±1.4 53.9 ±2.1	54.4 ±1.6 –	59.6 ±0.9 61.6 ±0.6	60.9 ±1.7 63.7 ±1.5	**61.1 ±1.2** **64.2 ±1.2**	52.9 ±1.0 52.0 ±1.3	55.5 ±1.0 55.0 ±1.2	53.0 ±1.1 52.5 ±1.6	53.9 ±1.6 53.5 ±1.6
Qwen3-1.7B	72.9 ±0.6	89.5 ±0.7	74.3 ±0.7 74.1 ±1.4	74.0 ±0.8 74.0 ±1.2	74.8 ±0.6 75.0 ±1.1	74.4 ±0.6 74.3 ±0.9	74.6 ±1.0 74.4 ±1.0	75.1 ±0.5 75.5 ±0.7	74.9 ±1.2 –	**77.9 ±0.7** **78.5 ±0.8**	77.6 ±0.4 78.1 ±0.8	74.5 ±0.9 74.5 ±1.0	74.7 ±0.9 74.7 ±0.9	74.2 ±1.1 73.8 ±1.4	74.2 ±1.1 74.2 ±1.3
Qwen3-4B	83.8 ±0.2	94.0 ±0.3	85.8 ±0.6 85.2 ±1.1	85.6 ±0.4 85.4 ±0.5	86.0 ±0.9 85.2 ±0.8	85.8 ±0.7 85.2 ±0.7	85.9 ±0.8 85.8 ±1.1	86.0 ±0.3 85.9 ±0.4	86.0 ±0.7 86.1 ±0.7	86.2 ±0.4 –	**86.8 ±0.6** **86.8 ±1.4**	85.8 ±0.9 85.7 ±1.0	85.9 ±0.5 85.5 ±0.5	85.3 ±0.9 85.0 ±0.9	85.7 ±0.4 85.2 ±0.9
Qwen3-8B	84.3 ±0.4	95.3 ±0.9	84.5 ±0.8 84.1 ±1.0	84.8 ±1.0 84.3 ±1.1	84.7 ±1.2 84.2 ±1.3	85.0 ±1.4 84.2 ±1.0	85.5 ±1.2 85.1 ±0.9	85.7 ±0.8 85.8 ±0.2	**86.6 ±0.7** 86.4 ±1.0	86.1 ±1.1 **87.0 ±0.6**	85.5 ±1.2 –	84.8 ±0.7 84.9 ±0.5	85.1 ±1.1 84.9 ±0.9	84.9 ±1.1 84.4 ±1.2	85.1 ±1.0 84.4 ±0.9
Gemma-3-1B	45.0 ±0.5	66.6 ±0.8	48.4 ±1.0 47.6 ±0.9	49.2 ±1.8 48.4 ±1.5	49.9 ±1.8 48.6 ±1.7	51.0 ±1.6 50.0 ±1.1	49.8 ±1.6 49.0 ±1.0	48.3 ±1.4 47.2 ±1.0	49.2 ±1.0 48.2 ±1.1	49.5 ±0.8 49.7 ±1.1	49.2 ±1.4 47.9 ±1.3	50.0 ±1.6 –	**53.2 ±0.7** **54.0 ±0.5**	48.9 ±1.6 48.4 ±0.9	49.3 ±2.3 48.9 ±1.3
Gemma-3-4B	74.3 ±0.3	88.5 ±0.8	78.2 ±0.8 78.4 ±1.0	78.1 ±0.6 78.2 ±0.4	77.6 ±0.7 78.4 ±0.6	78.2 ±0.5 **78.6 ±0.6**	78.2 ±0.4 78.5 ±0.7	78.0 ±0.6 77.6 ±0.7	78.4 ±0.6 78.0 ±0.7	78.3 ±0.6 78.1 ±1.1	**78.5 ±0.6** 78.2 ±0.5	**78.5 ±0.4** 77.9 ±0.5	78.4 ±0.4 –	**78.5 ±0.8** 78.3 ±1.0	78.3 ±0.5 78.4 ±0.9
Llama-3.2-1B	21.4 ±0.4	46.9 ±0.9	24.6 ±1.1 24.6 ±0.8	26.5 ±1.8 26.5 ±1.5	27.4 ±1.7 29.1 ±1.2	28.1 ±1.6 30.1 ±1.1	26.7 ±1.2 26.6 ±1.3	24.2 ±2.0 23.6 ±1.6	25.4 ±1.4 25.9 ±1.7	24.9 ±1.9 25.0 ±1.7	25.1 ±1.0 25.0 ±0.3	25.1 ±0.9 24.6 ±0.8	26.5 ±1.5 27.3 ±1.1	25.6 ±1.7 –	**28.4 ±1.9** **30.9 ±1.0**
Llama-3.2-3B	41.0 ±0.3	70.4 ±0.9	43.3 ±1.5 42.1 ±1.4	44.0 ±1.9 42.8 ±0.7	46.4 ±1.8 45.1 ±0.6	**47.0 ±0.8** **47.2 ±0.8**	45.7 ±1.2 44.8 ±1.5	43.9 ±0.7 43.1 ±1.2	44.6 ±1.4 44.4 ±1.1	43.7 ±1.4 43.1 ±1.1	45.0 ±1.1 43.3 ±1.4	43.1 ±0.8 42.4 ±0.6	44.3 ±0.4 44.8 ±0.4	44.2 ±0.7 42.5 ±0.8	45.4 ±0.9 –

Table 5: BoN@32 accuracy (%) on MATH-500 for every generator–scorer pair, selecting from all 32 samples. Layout and markings follow Table 4. Selection entries have no standard deviation because the 32 samples form a single group; pass@1 is the mean $\pm$ standard deviation over the 32 samples.

			Scored by Qwen2.5					Qwen3				Gemma 3		Llama 3.2	
Samples from	pass@1	pass@32	0.5B	1.5B	3B	7B	M-1.5B	0.6B	1.7B	4B	8B	1B	4B	1B	3B
Qwen2.5-0.5B	28.1 ±1.4	73.4	34.2 –	39.0 39.6	41.8 44.8	**43.0** **46.4**	41.8 45.0	33.0 31.8	38.0 38.6	39.0 39.6	37.4 37.0	34.2 33.4	37.4 38.0	33.4 31.8	38.4 37.8
Qwen2.5-1.5B	46.9 ±1.6	85.6	52.8 53.6	53.2 –	58.0 59.0	**59.2** **61.6**	57.8 61.0	48.2 48.0	53.4 53.8	54.4 54.4	54.2 54.0	51.8 50.8	52.4 52.6	49.0 46.8	51.2 50.6
Qwen2.5-3B	65.5 ±1.2	92.0	67.8 66.4	69.8 69.2	69.2 –	71.2 **72.8**	**71.8** 72.0	67.6 66.6	68.2 66.8	69.0 68.8	69.6 68.4	67.0 66.8	69.4 68.8	67.6 66.2	70.6 69.4
Qwen2.5-7B	75.8 ±1.1	94.2	78.4 77.6	78.2 77.0	78.8 79.4	79.6 –	**80.6** **80.4**	77.2 76.0	77.6 77.2	77.8 78.0	79.0 77.8	76.4 75.6	79.0 78.2	77.4 76.0	77.4 75.8
Q2.5-Math-1.5B	75.0 ±1.2	93.6	76.0 76.2	77.2 77.4	78.2 78.4	**80.0** **80.0**	78.2 –	75.8 75.4	76.6 75.8	76.2 75.6	75.4 75.8	74.4 73.6	77.2 76.8	75.4 75.6	76.2 75.8
Qwen3-0.6B	52.3 ±1.4	88.6	51.0 49.6	53.0 51.6	55.4 54.4	57.6 58.6	55.0 55.0	54.4 –	63.2 65.4	63.6 **69.4**	**64.2** 68.6	50.4 49.6	55.4 54.8	52.6 50.8	53.8 53.4
Qwen3-1.7B	72.9 ±1.4	94.6	74.4 73.4	73.8 73.8	74.6 74.4	73.8 73.4	74.8 73.6	75.4 75.2	75.8 –	78.4 **80.8**	**78.8** **80.8**	75.4 74.4	74.2 73.8	73.4 73.4	74.4 74.0
Qwen3-4B	83.8 ±0.9	96.4	86.0 85.2	85.6 83.8	85.6 85.0	86.2 84.8	85.8 84.6	85.6 85.8	87.2 86.6	86.8 –	**87.8** **87.2**	84.2 84.4	84.8 84.6	85.2 84.6	85.0 84.0
Qwen3-8B	84.3 ±1.0	97.8	83.8 82.8	84.4 84.0	84.0 83.8	84.8 84.0	85.4 85.2	**85.8** 85.6	**85.8** 86.2	**85.8** **87.4**	85.4 –	83.8 83.4	84.2 84.8	83.2 83.0	84.6 83.6
Gemma-3-1B	45.0 ±0.9	77.0	47.6 46.4	47.8 46.8	48.4 48.0	50.8 50.8	50.0 48.8	47.0 46.2	49.2 47.6	50.2 50.2	50.2 50.2	51.0 –	**55.6** **57.2**	49.4 48.0	48.6 48.4
Gemma-3-4B	74.3 ±1.0	93.0	79.2 79.0	79.0 **80.2**	79.0 78.8	79.4 79.6	80.0 79.4	78.2 78.4	79.2 79.4	79.2 79.0	79.6 79.4	79.4 79.0	79.8 –	79.2 79.4	**80.2** 79.6
Llama-3.2-1B	21.4 ±1.3	61.8	25.8 25.0	27.8 31.2	29.2 31.4	**31.0** 34.2	28.2 26.8	23.8 22.8	26.2 26.0	26.2 25.0	27.0 26.8	24.6 24.8	28.6 29.4	26.2 –	30.2 **34.8**
Llama-3.2-3B	41.0 ±1.5	83.4	42.0 41.2	43.8 42.2	46.6 47.8	**47.0** **48.4**	46.4 45.6	43.4 42.2	45.6 45.4	44.8 45.0	45.2 45.0	41.4 40.4	45.0 44.4	43.8 42.4	45.6 –

E ABLATION RESULTS

This appendix reports ablations of both stages of SMC-KD. Appendix E.1 studies teacher-guided SMC as a sampler: whether it maximizes the score that guides it, which potential to use, and how sensitive it is to the guidance strength. Appendix E.2 turns to the full training procedure, varying the potential used to construct the training data and the number of search rounds, and reports the fraction of responses that reach the token budget. Finally, Appendix E.3 evaluates the trained students when several responses are sampled per problem.

Table 6: Score (nats) of the response returned by BoN and SMC in the runs of Table 1, averaged over MATH-500 prompts: the self-likelihood $\log \pi_S$ for self-guided runs (top) and the ratio $\log \pi_T - \log \pi_S$ for teacher-guided runs (bottom). Entries are mean $\pm$ standard deviation over two seeds; bold marks the higher of BoN and SMC.

		$N = 4$		$N = 8$		$N = 16$		$N = 32$	
Samples from	Scored by	BoN	SMC	BoN	SMC	BoN	SMC	BoN	SMC
Qwen2.5-0.5B	Self	-68.5 ±0.5	**-64.3 ±0.4**	-58.9 ±0.6	**-52.5 ±0.7**	-52.0 ±0.6	**-45.2 ±0.9**	-46.2 ±0.1	**-40.7 ±0.6**
	Qwen2.5-Math-1.5B	-78.7 ±0.4	**-66.1 ±0.9**	-66.5 ±0.4	**-48.6 ±1.2**	-57.6 ±0.5	**-37.5 ±0.0**	-49.7 ±0.2	**-28.6 ±0.7**
Qwen3-0.6B	Self	**-70.6 ±0.7**	-71.4 ±1.1	**-60.7 ±0.2**	-63.6 ±0.8	**-54.2 ±0.5**	-58.0 ±0.6	**-49.3 ±0.1**	-53.2 ±0.4
	Qwen3-1.7B	-49.5 ±0.2	**-31.6 ±0.2**	-40.6 ±0.1	**-17.2 ±0.9**	-34.4 ±0.1	**-8.4 ±0.6**	-29.4 ±0.3	**-0.9 ±0.1**
Gemma-3-1B	Self	-82.9 ±1.7	**-71.8 ±1.5**	-68.2 ±0.8	**-56.3 ±2.2**	-58.6 ±0.5	**-44.1 ±2.6**	-50.7 ±1.0	**-37.7 ±0.4**
	Gemma-3-4B	-132.5 ±2.9	**-103.3 ±1.5**	-112.9 ±1.7	**-75.7 ±1.3**	-97.4 ±0.4	**-53.1 ±0.0**	-84.5 ±0.9	**-38.6 ±0.0**
Llama-3.2-1B	Self	-72.1 ±0.5	**-71.7 ±0.8**	-63.0 ±0.4	**-62.0 ±0.3**	-56.7 ±0.1	**-54.9 ±1.2**	-52.0 ±0.2	**-51.0 ±0.3**
	Llama-3.2-3B	-28.5 ±0.9	**-21.8 ±0.5**	-21.5 ±0.0	**-13.6 ±0.8**	-16.8 ±0.1	**-7.6 ±0.2**	-13.1 ±0.0	**-3.0 ±0.1**

Table 7: SMC@8 accuracy (%) on MATH-500 for Qwen2.5-0.5B guided by Qwen2.5-Math-1.5B, for three potentials and five guidance strengths λ. BoN@8 applies the same potential to 8 independent samples (from Table 4). SMC entries are mean $\pm$ standard deviation over two seeds; bold marks the best entry in each row.

Score (V)	BoN@8	SMC@8				
		$\lambda = 0.5$	$\lambda = 0.75$	$\lambda = 1$	$\lambda = 1.5$	$\lambda = 2$
Self ($\log \pi_S$)	34.6 ±0.6	34.7 ±0.1	35.0 ±0.8	**35.2 ±0.6**	33.2 ±0.3	34.4 ±0.6
Teacher ($\log \pi_T$)	38.3 ±1.4	43.7 ±1.8	44.4 ±1.4	44.6 ±1.1	**45.0 ±0.3**	44.3 ±1.3
Ratio ($\log(\pi_T/\pi_S)$)	39.9 ±0.9	46.0 ±2.8	45.8 ±0.8	43.9 ±1.3	**46.6 ±0.6**	44.8 ±0.6

E.1 TEACHER-GUIDED SMC

All runs in this section use the direct-generation setting of Section 4.2, with 64-token blocks and resampling whenever ESS $< N/2$ (Appendix C). We vary the guidance strength λ by raising each incremental weight w_h^i in Algorithm 1 to the power λ, which maps (α, β) to $(\lambda\alpha, \lambda\beta)$ in Equation 4. For the ratio potential, the target becomes proportional to $\bar{\pi}_S^{1-\lambda}\pi_T^{\lambda}$: $\lambda < 1$ interpolates between the student and the teacher, $\lambda = 1$ recovers π_T under the support condition of Section 3.3, and $\lambda > 1$ extrapolates beyond the teacher. All other experiments use $\lambda = 1$. The BoN baselines in Tables 7–9 are computed from the 32 samples per prompt of Appendix D, split into $32/N$ disjoint groups of N, rather than from the two independent BoN runs of Table 1; the two estimates differ by at most 1.1 points.

We first check that SMC maximizes the score that guides it. Table 6 reports, for the runs of Table 1, the score of the response that each method returns. Because the scores are summed log-likelihoods, they are comparable across methods and particle counts for the same generator but not across generators. Under teacher guidance, SMC attains a higher log-likelihood ratio than BoN for every pairing and particle count, and its advantage widens as N grows from 4 to 32: from 12.6 to 21.1 nats for Qwen2.5-0.5B, from 17.9 to 28.5 nats for Qwen3-0.6B, from 29.2 to 45.9 nats for Gemma-3-1B, and from 6.7 to 10.1 nats for Llama-3.2-1B. Under self-guidance, SMC attains a higher self-likelihood than BoN in 12 of the 16 settings; the exception is Qwen3-0.6B, for which BoN is higher at every N. Both methods reach higher scores as N increases in every setting.

A higher score is useful only if it also means more correct answers, and comparing Table 6 with Table 1 shows that this holds for the ratio but not for the self-likelihood. Under teacher guidance, SMC attains both the higher ratio and the higher accuracy in all 16 settings, and its accuracy increases with N in every pairing except Gemma-3-1B at $N = 32$. Under self-guidance, score and accuracy diverge. For Gemma-3-1B, SMC attains a higher self-likelihood than BoN at every N, by 11 to 15 nats, yet returns fewer correct answers at every N, for example 47.1% versus 51.6% at $N = 32$. As its score improves from -71.8 to -37.7 nats, its accuracy stays within 2.3 points of pass@1

Table 8: SMC@N accuracy (%) on MATH-500 across guidance strengths λ and particle counts N, for Qwen2.5-0.5B guided by Qwen2.5-Math-1.5B with the ratio potential. BoN@N is the mean $\pm$ standard deviation over $32/N$ disjoint groups of samples, and SMC over two seeds. Bold marks the best entry in each row.

N	BoN@N	SMC@N				
		$\lambda = 0.5$	$\lambda = 0.75$	$\lambda = 1$	$\lambda = 1.5$	$\lambda = 2$
4	36.9 ±1.0	41.4 ±0.6	**41.6 ±0.3**	40.0 ±1.4	39.5 ±2.4	41.2 ±1.1
8	39.9 ±0.9	46.0 ±2.8	45.8 ±0.8	43.9 ±1.3	**46.6 ±0.6**	44.8 ±0.6
16	42.7 ±2.1	**49.8 ±2.0**	48.2 ±0.6	47.7 ±0.4	47.8 ±1.7	49.0 ±0.3
32	45.0	**53.4 ±2.0**	53.1 ±1.8	**53.4 ±1.1**	52.2 ±0.6	51.9 ±2.4

Table 9: SMC@8 accuracy (%) on MATH-500 across guidance strengths λ for the four pairings of Table 1, with the ratio potential. BoN@8 is from Table 4, and SMC entries are mean $\pm$ standard deviation over two seeds. Bold marks the best entry in each row.

Samples from	Scored by	BoN@8	SMC@8				
			$\lambda = 0.5$	$\lambda = 0.75$	$\lambda = 1$	$\lambda = 1.5$	$\lambda = 2$
Qwen2.5-0.5B	Qwen2.5-Math-1.5B	39.9 ±0.9	46.0 ±2.8	45.8 ±0.8	43.9 ±1.3	**46.6 ±0.6**	44.8 ±0.6
Qwen3-0.6B	Qwen3-1.7B	61.6 ±0.6	64.8 ±0.6	64.5 ±1.3	**65.3 ±0.4**	64.3 ±1.8	63.6 ±0.3
Gemma-3-1B	Gemma-3-4B	54.0 ±0.5	56.3 ±0.1	57.8 ±0.3	58.6 ±2.3	**58.9 ±0.4**	58.3 ±1.0
Llama-3.2-1B	Llama-3.2-3B	30.9 ±1.0	33.6 ±1.4	33.8 ±2.3	33.1 ±2.4	**34.6 ±0.0**	31.9 ±0.4

(45.3%). Conversely, for Qwen3-0.6B, self-guided SMC attains the lower score but matches or exceeds BoN in accuracy. Pushing the self-likelihood higher therefore does not make answers more likely to be correct, whereas pushing the ratio higher does. This supports guiding SMC-KD with the ratio rather than with the student's own likelihood.

Table 7 isolates the role of the teacher by comparing the self, teacher-only, and ratio potentials directly, for Qwen2.5-0.5B guided by Qwen2.5-Math-1.5B at five values of λ. Both teacher-based potentials outperform self-guidance by 8.7 to 13.4 points at every λ. With either one, SMC beats BoN with the same score at every λ, by 5.4 to 6.7 points for $\log \pi_T$ and by 4.0 to 6.7 points for the ratio, whereas self-guided SMC stays within 1.4 points of self-scored BoN. The ratio gives the highest accuracy at four of the five values of λ; at $\lambda = 1$, the teacher-only potential is higher by 0.7 points, within one standard deviation.

Compared with the choice of potential, the guidance strength matters little. Tables 8 and 9 vary λ for the ratio potential across particle counts and pairings, and no value is consistently best. For Qwen2.5-0.5B, the best value changes with the particle count, from $\lambda = 0.75$ at $N = 4$ to 1.5 at $N = 8$ and 0.5 at $N = 16$, with 0.5 and 1 tied at $N = 32$. At $N = 8$, $\lambda = 1.5$ has the best mean in three of the four pairings, but for Qwen2.5-0.5B, the one pairing that we also sweep over N, it is the worst value at $N = 4$ and the second worst at $N = 32$. In every row of Tables 7 to 9, accuracy varies by at most 2.7 points across $\lambda \in [0.5, 2]$, comparable to the standard deviation across seeds, which reaches 2.8 points. The particle count matters far more: from $N = 4$ to $N = 32$, accuracy increases by 10.7 to 13.4 points at every λ. Every value of λ also beats BoN at every particle count and in every pairing, and the margin of the worst λ over BoN grows from 2.6 points at $N = 4$ to 6.9 points at $N = 32$. We therefore use $\lambda = 1$ throughout without tuning. It is the value at which the ratio-guided target equals π_T, and it is within 2.7 points of the best λ in every row of Tables 8 and 9.

E.2 SMC-KD

For training, the choice of potential interacts with the number of search rounds. Table 10 trains SMC-KD students with the teacher-only and ratio potentials, each with a single search round ($K = 1$) or twelve rounds ($K = 12$), using the same number of optimizer updates. The teacher-only potential

Table 10: MATH-500 pass@1 (%) of SMC-KD students trained with the teacher-only ($\log \pi_T$) or ratio ($\log \pi_T - \log \pi_S$) potential and K search rounds, with $N = 8$ and 750 optimizer updates in every arm. Entries are mean $\pm$ standard deviation over three training seeds for the ratio and two for the teacher-only potential. Bold marks the best arm in each pairing.

Student ← Teacher	Teacher ($K = 1$)	Teacher ($K = 12$)	Ratio ($K = 1$)	Ratio ($K = 12$)
Qwen2.5-1.5B ← Qwen2.5-Math-1.5B	52.8 ±0.1	55.6 ±0.2	52.7 ±0.1	**55.9 ±0.3**
Qwen2.5-0.5B ← Qwen2.5-Math-1.5B	32.1 ±0.2	31.3 ±0.6	**32.4 ±0.4**	32.2 ±0.5
Qwen2.5-0.5B ← Qwen3-1.7B	**31.3 ±1.1**	29.6 ±0.4	31.1 ±0.1	30.7 ±0.5

Table 11: Accuracy (%) of the base student, OPD, and both SMC-KD variants, with the percentage of responses that reach the 2,048-token budget without finishing in parentheses. SMC-KD$_{12}$ is iterative ($K = 12$) and SMC-KD$_1$ one-shot ($K = 1$), both with the ratio potential. Trained rows are mean $\pm$ standard deviation over three training seeds, and the base row over sampling seeds. Bold marks the best method per benchmark in each pairing.

Student ← Teacher	Method	MATH-500	GSM8K	AMC23
Qwen2.5-1.5B ← Qwen2.5-Math-1.5B	Base	46.9 ±1.6 (0.7%)	65.4 ±3.1 (0.0%)	23.1 ±1.1 (1.9%)
	OPD	54.0 ±0.1 (1.3%)	72.9 ±0.5 (0.1%)	25.2 ±1.3 (2.5%)
	SMC-KD$_{12}$	**55.9 ±0.3** (3.1%)	**73.9 ±1.1** (0.2%)	26.0 ±1.0 (6.4%)
	SMC-KD$_1$	52.7 ±0.1 (1.6%)	73.5 ±0.5 (0.1%)	**27.5 ±2.2** (2.7%)
Llama-3.2-1B ← Llama-3.2-3B	Base	21.4 ±1.3 (8.6%)	37.1 ±1.6 (0.4%)	10.1 ±0.4 (13.9%)
	OPD	24.2 ±0.4 (12.4%)	38.8 ±1.1 (0.9%)	11.4 ±0.5 (19.3%)
	SMC-KD$_{12}$	**24.8 ±0.8** (15.6%)	**43.5 ±1.0** (1.7%)	12.9 ±2.3 (26.5%)
	SMC-KD$_1$	24.3 ±0.5 (13.0%)	40.2 ±0.3 (1.5%)	**13.1 ±1.1** (23.5%)
Qwen2.5-0.5B ← Qwen2.5-Math-1.5B	Base	28.1 ±1.4 (1.7%)	40.8 ±1.7 (0.2%)	10.2 ±0.5 (3.3%)
	OPD	32.0 ±0.2 (3.6%)	**47.7 ±0.5** (0.4%)	**13.9 ±0.8** (7.6%)
	SMC-KD$_{12}$	32.2 ±0.5 (12.2%)	47.5 ±1.1 (0.8%)	11.7 ±2.3 (18.6%)
	SMC-KD$_1$	**32.4 ±0.4** (5.5%)	46.5 ±0.6 (0.4%)	12.7 ±0.4 (9.4%)
Qwen2.5-0.5B ← Qwen3-1.7B	Base	28.1 ±1.4 (1.7%)	40.8 ±1.7 (0.2%)	**10.2 ±0.5** (3.3%)
	OPD	29.9 ±0.3 (5.5%)	43.1 ±0.3 (1.2%)	9.1 ±1.2 (11.4%)
	SMC-KD$_{12}$	30.7 ±0.5 (16.0%)	**47.1 ±0.3** (0.7%)	6.1 ±1.5 (23.5%)
	SMC-KD$_1$	**31.1 ±0.1** (5.3%)	46.0 ±0.1 (0.5%)	**10.2 ±1.5** (10.4%)

also improves on the base student in all six settings, but the ratio gives the best arm in two of the three pairings. With a single round, the two potentials are indistinguishable, differing by at most 0.3 points. With twelve rounds, the ratio is better in all three pairings, by 0.3, 0.9, and 1.1 points. This is consistent with Section 3.3, where the ratio-guided target equals π_T under a support condition, regardless of the current student. The teacher-only potential gives the best arm only for the cross-family pairing Qwen2.5-0.5B ← Qwen3-1.7B at $K = 1$, by 0.2 points, within one standard deviation. This agrees with Appendix D, where the ratio's advantage over the teacher's log-likelihood is concentrated in same-family pairs.

Table 11 extends the comparison of $K = 1$ and $K = 12$ to all four pairings and all three benchmarks. Table 2 uses $K = 12$ for Qwen2.5-1.5B and Llama-3.2-1B and $K = 1$ for the two Qwen2.5-0.5B pairings; with either value of K fixed for all four pairings, SMC-KD still outperforms OPD in 9 of the 12 pairing–benchmark comparisons. Repeating the search does have a cost, visible in the percentage of responses that reach the 2,048-token budget without finishing: iterating increases this capped fraction for every pairing and benchmark, and the increase is concentrated in the Qwen2.5-0.5B student. On MATH-500, its capped fraction grows from 5.5% to 12.2% with Qwen2.5-Math-

1.5B and from 5.3% to 16.0% with Qwen3-1.7B, compared with 1.6% to 3.1% for Qwen2.5-1.5B and 13.0% to 15.6% for Llama-3.2-1B. Because each round trains the student on responses it generated itself, degenerate patterns such as repeated reasoning can be reinforced in later rounds once they enter the training data. How much this costs in accuracy depends on the teacher. With Qwen2.5-Math-1.5B, the iterative student stays within 0.2 points of the one-shot student on MATH-500 and within 1.0 point on AMC23. With Qwen3-1.7B, its AMC23 accuracy falls from 10.2% to 6.1%, below the base student, and 23.5% of its AMC23 responses are capped. On GSM8K, where fewer than 1% of this student's SMC-KD responses are capped, iteration improves accuracy slightly with both teachers. This suggests that the math-specialized teacher separates useful trajectories from the student's degenerate ones better than the cross-family teacher, consistent with its higher selection accuracy for this student in Appendix D.

E.3 Multi-sample coverage and response diversity

Table 12 evaluates the trained students when more than one response is sampled per problem. For the base student, BoN@8 is computed from the same 32-sample pool as Table 4. An SMC-KD variant has the best pass@1 in every pairing, but this advantage does not carry over to the multi-sample metrics. OPD has the highest BoN@8 in three of the four pairings, and the best pass@8 is split between OPD and SMC-KD. The iterative variant is affected most. SMC-KD$_{12}$ lowers pass@8 below that of the base student in three of the four pairings, and in every pairing it has the smallest gap between pass@8 and pass@1, so its additional samples add the least beyond the first. The effect is most pronounced for Llama-3.2-1B, where SMC-KD$_{12}$ has the highest pass@1 (24.8%) but the lowest pass@8 (44.0%, compared with 53.5% for OPD and 46.9% for the base student). We attribute this to reduced response diversity. SMC-KD trains the student on the output of teacher-guided search, and resampling concentrates each population on a few high-weight prefixes; repeating the search with an already concentrated student compounds the effect. The one-shot variant keeps pass@8 above the base student in every pairing, which makes it the safer choice when the student will be used with repeated sampling. Both variants also gain less than OPD from inference-time selection in every pairing: BoN@8 exceeds pass@1 by 2.2 to 4.7 points for SMC-KD and by 5.0 to 5.6 points for OPD. This is consistent with part of the selection gain having already been distilled into the student's single-sample accuracy.

Table 12: Multi-sample metrics on MATH-500 with 8 samples per problem, for the models of Table 11. BoN@8 and SMC@8 select one of the 8 samples by the model's own log-likelihood, without the teacher. Trained rows are mean ± standard deviation over three training seeds, and the base row over sampling seeds. Bold marks the best method per column in each pairing.

Student ← Teacher	Method	pass@1	pass@8	BoN@8	SMC@8
Qwen2.5-1.5B ← Qwen2.5-Math-1.5B	Base	46.9 ±1.6	75.0 ±1.5	52.8 ±2.8	53.4
	OPD	54.0 ±0.1	77.0 ±0.9	**59.6 ±1.0**	58.3 ±1.4
	SMC-KD$_{12}$	**55.9 ±0.3**	**77.1 ±1.6**	58.1 ±0.6	**58.5 ±0.5**
	SMC-KD$_1$	52.7 ±0.1	77.0 ±0.4	56.7 ±0.9	55.9 ±0.2
Llama-3.2-1B ← Llama-3.2-3B	Base	21.4 ±1.3	46.9 ±0.9	25.6 ±1.7	25.8
	OPD	24.2 ±0.4	**53.5 ±0.6**	**29.7 ±0.8**	**27.6 ±1.3**
	SMC-KD$_{12}$	**24.8 ±0.8**	44.0 ±1.7	28.4 ±1.7	27.1 ±2.2
	SMC-KD$_1$	24.3 ±0.5	48.9 ±1.1	27.0 ±0.5	26.6 ±0.7
Qwen2.5-0.5B ← Qwen2.5-Math-1.5B	Base	28.1 ±1.4	57.5 ±0.7	34.6 ±0.6	34.8
	OPD	32.0 ±0.2	**59.7 ±2.0**	**37.4 ±1.2**	36.9 ±0.4
	SMC-KD$_{12}$	32.2 ±0.5	55.3 ±1.2	34.8 ±0.9	37.1 ±0.3
	SMC-KD$_1$	**32.4 ±0.4**	59.5 ±1.8	37.0 ±1.1	**37.2 ±1.2**
Qwen2.5-0.5B ← Qwen3-1.7B	Base	28.1 ±1.4	57.5 ±0.7	34.6 ±0.6	34.8
	OPD	29.9 ±0.3	56.5 ±1.0	34.9 ±0.5	34.1 ±1.1
	SMC-KD$_{12}$	30.7 ±0.5	52.5 ±1.1	32.9 ±1.0	30.7 ±0.3
	SMC-KD$_1$	**31.1 ±0.1**	**59.2 ±1.1**	**35.8 ±0.5**	**35.6 ±0.9**