IOD: AN ITERATIVE ON-POLICY DISTILLATION FRAMEWORK FOR SELF-IMPROVING LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy knowledge distillation reduces the mismatch between student training and inference by learning from student-generated trajectories. However, extending distillation across multiple student updates raises the question of how previously generated trajectories should be managed as the student evolves. We propose Iterative On-Policy Distillation (IOD), a multi-cycle framework that combines teacher-based trajectory filtering with cross-cycle accumulation. At each cycle, IOD jointly re-evaluates previously retained and newly generated trajectories using a fixed teacher, then updates the student through token-level distribution matching while preserving the original human-labeled data. Experiments on summarization and Machine translation, using encoder–decoder and decoder-only teacher–student architectures, show higher task-specific scores than reproduced distillation baselines under the evaluated settings. Ablation studies further examine the effects of trajectory filtering, accumulation, and the student-update objective. These findings highlight trajectory management as an important consideration when extending on-policy distillation across successive student updates.

1 INTRODUCTION

Large language models (LLMs) have demonstrated strong natural language generation capabilities, but their computational and memory requirements hinder deployment in resource-constrained environments. Knowledge distillation (KD) transfers the predictive behavior of a large teacher to a smaller student model (Hinton et al., 2015; Kim & Rush, 2016). However, autoregressive distillation suffers from a train–test mismatch: teacher-forced training conditions on reference prefixes, whereas inference conditions on the student's own predictions. This discrepancy, commonly associated with exposure bias, can lead to error accumulation during generation (Bengio et al., 2015). On-policy distillation addresses this mismatch by training on student-generated trajectories. Generalized Knowledge Distillation (GKD) (Agarwal et al., 2024), for example, matches teacher and student next-token distributions under student-generated prefixes.

A challenge arises when on-policy distillation is extended across successive student updates: the distribution of student-generated trajectories changes as the student evolves, requiring decisions about which trajectories to retain and reuse. Existing approaches address related aspects of trajectory construction and generated-data reuse. Speculative Knowledge Distillation (SKD) (Xu et al., 2025) incorporates teacher feedback during generation, while Reinforced Self-Training (ReST) (Gulcehre et al., 2023) iteratively generates, evaluates, and reuses model-generated data. Other studies investigate distillation objectives, sampling strategies, and generated-data quality (Ko et al., 2024; Wang et al., 2023). In multi-cycle distillation, relying only on current-cycle trajectories discards potentially useful earlier contexts, whereas retaining all past generations can preserve low-quality samples and increase training costs. Permanently retaining previously selected trajectories also prevents newer candidates from replacing them. These considerations motivate a cross-cycle mechanism that jointly evaluates previously retained and newly generated trajectories under a consistent selection criterion.

We propose Iterative On-policy Distillation (IOD), a multi-cycle framework that constructs an evolving teacher-guided curriculum through a Grow–Evaluate/Filter–Improve procedure. At each cycle, the student generates candidate trajectories, which a fixed teacher evaluates using length-normalized

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.

conditional log-likelihood. Previously retained trajectories are combined with new candidates and jointly re-filtered, allowing the retained set to evolve without unconditionally accumulating all past generations. The student is then updated through token-level teacher–student distribution matching on the selected trajectories and the original human-annotated data, which remain available throughout training.

We focuses on cross-cycle selection and reuse rather than introducing a new distillation objective or an additional reward model. Unlike methods that involve the teacher during generation, IOD generates candidates using the student alone and uses the teacher for trajectory evaluation and distribution-level supervision. Teacher likelihood provides a consistent selection criterion but does not guarantee downstream task quality. Furthermore, trajectories retained from earlier cycles are not necessarily on-policy with respect to the current student. We evaluate IOD on summarization and machine translation using encoder–decoder and decoder-only models. Through controlled comparisons with distillation baselines under matched experimental settings, we assess IOD across different tasks and model architectures. We further investigate the effects of trajectory filtering, cross-cycle accumulation, student-update objectives, and retention-ratio selection through ablation studies.

Our main contributions are summarized as follows:

- We formulate cross-cycle trajectory selection and reuse as an explicit challenge in multi-cycle on-policy distillation, addressing the management of student-generated trajectories as the student policy evolves.
- We propose IOD, which constructs an evolving teacher-guided curriculum by jointly re-selecting previously retained and newly generated student trajectories while preserving the original human-annotated data.
- We conduct controlled experiments on summarization and machine translation using encoder–decoder and decoder-only models, comparing IOD with reproduced distillation baselines under matched experimental settings and analyzing its key components through ablation studies.

2 Related Work

On-Policy Knowledge Distillation Knowledge distillation transfers predictive knowledge from a teacher to a smaller student through distribution matching or sequence-level supervision (Hinton et al., 2015; Kim & Rush, 2016). To mitigate the mismatch between teacher-forced training and autoregressive inference, ImitKD (Lin et al., 2020) and Generalized Knowledge Distillation (GKD) (Agarwal et al., 2024) incorporate student-generated prefixes into teacher-supervised learning. Recent methods further investigate the objectives and data-utilization strategies for generative distillation. MiniLLM (Gu et al., 2024) employs reverse-KL optimization, while DistiLLM (Ko et al., 2024) combines skew-KL divergence with adaptive off-policy data utilization. DistiLLM-2 (Ko et al., 2025) introduces contrastive distillation using teacher- and student-generated responses. Speculative Knowledge Distillation (SKD) (Xu et al., 2025) addresses the quality of student-generated trajectories through teacher intervention during token generation. While these approaches investigate student-conditioned supervision, distillation objectives, and trajectory construction, IOD focuses on the selection and reuse of student-generated trajectories across successive model updates.

Iterative Self-Training with Model-Generated Data Iterative self-training improves models through repeated generation, evaluation, and learning from model-generated data. STaR (Zelikman et al., 2022) applies this principle to reasoning, while Reinforced Self-Training (ReST) (Gulcehre et al., 2023) employs reward-guided data selection and supports the reuse of generated samples across policy-improvement stages. ReSTEM (Singh et al., 2024) similarly combines repeated generation, feedback-based filtering, and model updates. Related approaches, such as Self-Play Fine-Tuning (SPIN) (Chen et al., 2024b), further investigate iterative improvement using responses generated by earlier model iterations. These methods establish that repeated generation and data reuse are not sufficient to distinguish IOD from existing self-training approaches. IOD considers a teacher–student distillation setting in which the selection of training trajectories and the distribution-matching objective serve distinct roles, with particular attention to the reuse of trajectories generated across successive student updates.

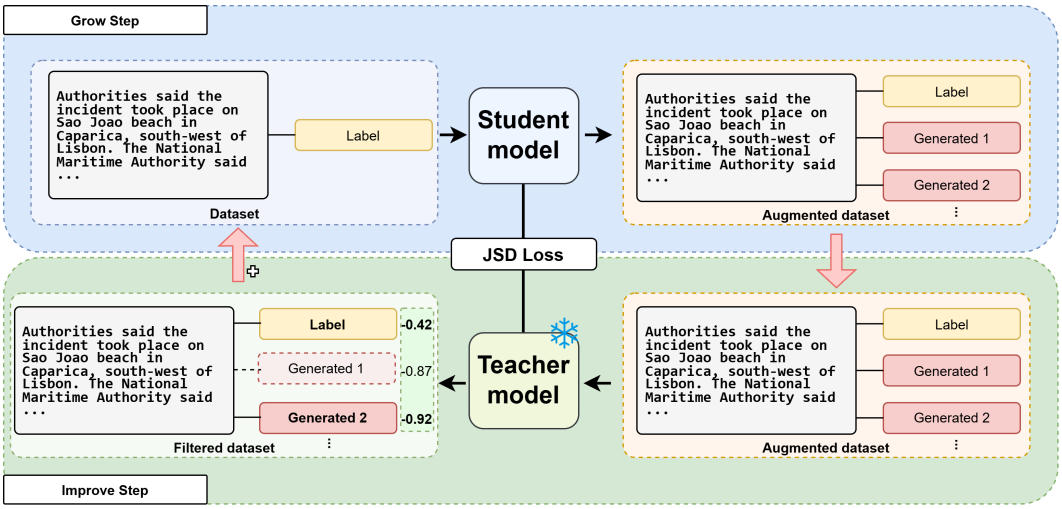

Figure 1: Overview of Iterative On-policy Distillation (IOD). At each training cycle, the student generates candidate outputs from the original training inputs (Grow). A fixed teacher evaluates the completed student-generated outputs using their average token-level log-likelihood and selects a high-scoring subset (Evaluate/Filter). Previously retained outputs are reconsidered together with newly generated candidates, allowing the selected training data to evolve across cycles. The student is then updated by minimizing the Jensen–Shannon divergence between teacher and student next-token distributions on the selected outputs and the original human-annotated data, which are preserved throughout training (Improve).

Generated-Data Selection and Curriculum Construction The quality and composition of model-generated data play an important role in language-model training. Self-Instruct (Wang et al., 2023) and AlpaGasus (Chen et al., 2024a) investigate filtering strategies for constructing synthetic instruction-tuning datasets, while DEITA (Liu et al., 2024) considers data quality, complexity, and diversity in data selection. LESS (Xia et al., 2024) further investigates training-example selection based on estimated influence on downstream tasks. These studies demonstrate that selecting training examples involves considerations beyond the quantity of available generated data. In iterative learning, data selection must also account for changes in the model's generation behavior across successive updates. IOD examines this issue in the context of teacher–student distillation, focusing on the selection and reuse of student-generated trajectories collected at different stages of training.

3 Methodology

We present Iterative On-policy Distillation (IOD), a multi-cycle framework that constructs an evolving training curriculum from student-generated trajectories. IOD separates student-side candidate generation, teacher-based trajectory selection, and token-level distillation. The central operation is cross-cycle trajectory management, which jointly considers previously retained and newly generated trajectories when constructing the training set for each student update. Figure 1 illustrates the overall framework.

3.1 Problem Setup

Let π_T denote a fixed teacher model and $\pi_S^{(c)}$ denote the student model after distillation cycle $c \in \{0, 1, \ldots, C\}$, where C is the total number of distillation cycles. The initial student $\pi_S^{(0)}$ is obtained through supervised fine-tuning on the original human-labeled dataset, defined as

$$\mathcal{D}_H = \left\{(x_i, y_i^H)\right\}_{i=1}^{N}, \tag{1}$$

where x_i denotes an input and y_i^H is its corresponding human-written reference. The original input set used for student generation is then given by

$$\mathcal{X}_H = \left\{x_i : (x_i, y_i^H) \in \mathcal{D}_H\right\}. \tag{2}$$

The human-labeled dataset $\mathcal{D}_H$ is preserved throughout all distillation cycles, while teacher-based filtering is applied exclusively to student-generated trajectories.

At each cycle $c \in \{1, \ldots, C\}$, the student model from the previous cycle, $\pi_S^{(c-1)}$, generates K candidate outputs for each input $x_i \in \mathcal{X}_H$, as follows:

$$\hat{y}_{i,k}^{(c)} \sim \pi_S^{(c-1)}(\cdot \mid x_i), \qquad k = 1, \ldots, K. \tag{3}$$

These generated outputs form the candidate pool for cycle c, denoted by

$$\mathcal{P}^{(c)} = \left\{ (x_i, \hat{y}_{i,k}^{(c)}) \right\}_{i=1,k=1}^{N,K}. \tag{4}$$

Each trajectory in $\mathcal{P}^{(c)}$ is sampled from the student policy $\pi_S^{(c-1)}$ that generated it. The resulting candidate pool serves as the source of newly generated trajectories for teacher-based evaluation and subsequent curriculum construction.

3.2 Teacher-based Trajectory Evaluation and Filtering

Teacher-based trajectory score. To evaluate student-generated trajectories, the fixed teacher assigns a length-normalized conditional log-likelihood to each input–output pair (x, y), where $y = (y_1, \ldots, y_{|y|})$. The teacher score is defined as follows:

$$s_T(x, y) = \frac{1}{|y|} \sum_{t=1}^{|y|} \log \pi_T(y_t \mid x, y_{<t}), \tag{5}$$

where $|y|$ denotes the number of valid target tokens, excluding padding. Length normalization reduces the direct influence of sequence length when comparing generated trajectories.

Since the teacher remains fixed throughout training, $s_T(x, y)$ provides a consistent criterion for ranking trajectories generated at different distillation cycles. The score reflects the likelihood assigned by the teacher to a completed student-generated sequence, rather than a guarantee of downstream task quality or lower teacher–student divergence. The teacher is used to evaluate generated trajectories and does not participate in student-side candidate generation.

Top-q filtering. Based on the teacher scores, IOD selects student-generated trajectories for subsequent training. Given a candidate multiset $\widetilde{\mathcal{A}}$ and a retention ratio $q \in (0, 1]$, let τ_q denote the empirical $(1-q)$-quantile of its teacher scores, computed using linear interpolation. The filtering operation is defined as

$$\mathrm{TopQ}(\widetilde{\mathcal{A}}, s_T, q) = \left\{ z \in \widetilde{\mathcal{A}} : s_T(z) \geq \tau_q \right\}_{\mathrm{multi}}. \tag{6}$$

Here, $\{\cdot\}_{\mathrm{multi}}$ denotes the submultiset that preserves individual trajectory occurrences satisfying the selection condition. All trajectory occurrences satisfying the threshold are retained, including those tied at the cutoff. Consequently, the realized retention count need not equal the nominal top-q count. Selection is performed globally over the student-generated collection rather than independently for each input. The resulting multiset forms the retained student-generated curriculum.

3.3 Cross-cycle Trajectory Management

IOD constructs an evolving training curriculum by selecting student-generated trajectories across successive distillation cycles. Let $\mathcal{A}^{(c)}$ denote the collection of student-generated trajectories retained at cycle c. Before the first cycle, the retained collection is initialized as empty:

$$\mathcal{A}^{(0)} = \varnothing. \tag{7}$$

At cycle c, IOD combines the trajectories retained from the previous cycle with the newly generated candidate pool to construct a joint candidate collection, as follows:

$$\widetilde{\mathcal{A}}^{(c)} = \mathcal{A}^{(c-1)} \uplus \mathcal{P}^{(c)}. \tag{8}$$

Here, $\uplus$ denotes multiset concatenation, which preserves individual trajectory occurrences. The retained curriculum is then updated by applying the teacher-based filtering operation defined in Eq. 6 to the combined collection:

$$\mathcal{A}^{(c)} = \mathrm{TopQ}\left(\widetilde{\mathcal{A}}^{(c)}, s_T, q\right). \tag{9}$$

This joint selection allows previously retained and newly generated trajectories to be evaluated under the same fixed teacher criterion. Unlike unconditional accumulation, previously retained trajectories are not automatically preserved in subsequent cycles; they remain in the curriculum only if they satisfy the updated selection criterion. Consequently, newly generated candidates can replace previously retained trajectories during curriculum construction.

The distinction between newly generated and retained trajectories is important when interpreting the on-policy nature of IOD. Each trajectory in $\mathcal{P}^{(c)}$ is sampled from the student policy $\pi_S^{(c-1)}$ used for generation, whereas trajectories carried over from $\mathcal{A}^{(c-1)}$ may originate from earlier student checkpoints. Therefore, the retained curriculum $\mathcal{A}^{(c)}$ contains trajectories generated under different student policies and is not necessarily entirely on-policy with respect to the current student.

Finally, IOD combines the retained student-generated curriculum with the original human-labeled dataset to construct the training dataset for cycle c, defined as follows:

$$\mathcal{D}_{\mathrm{train}}^{(c)} = \mathcal{D}_H \uplus \mathcal{A}^{(c)}. \tag{10}$$

The original human-labeled data are preserved throughout training without teacher-based filtering, while the student-generated component is updated through cross-cycle trajectory selection.

3.4 Student Update and Overall Procedure

After constructing $\mathcal{D}_{\mathrm{train}}^{(c)}$, the student is updated through token-level teacher–student distribution matching. While the teacher score in Eq. 5 is used for trajectory selection, the student-update objective compares the complete next-token distributions of the teacher and student.

Let π_θ denote the trainable student initialized from $\pi_S^{(c-1)}$ at the beginning of cycle c. For each training trajectory $(x, y) \in \mathcal{D}_{\mathrm{train}}^{(c)}$, we denote the teacher and student next-token distributions at position t by p_t and q_t, respectively, as follows:

$$p_t = \pi_T(\cdot \mid x, y_{<t}), \qquad q_t = \pi_\theta(\cdot \mid x, y_{<t}). \tag{11}$$

We measure the discrepancy between the teacher and student distributions using the Jensen–Shannon divergence (JSD), defined as follows:

$$\begin{aligned} D_{\mathrm{JSD}}(p_t \| q_t) &= \frac{1}{2} D_{\mathrm{KL}}(p_t \| m_t) + \frac{1}{2} D_{\mathrm{KL}}(q_t \| m_t), \\ m_t &= \frac{1}{2}(p_t + q_t). \end{aligned} \tag{12}$$

The length-normalized distillation loss for each training trajectory is then defined as follows:

$$\ell_{\mathrm{JSD}}(\theta; x, y) = \frac{1}{|y|} \sum_{t=1}^{|y|} D_{\mathrm{JSD}}(p_t \| q_t). \tag{13}$$

At each cycle, the student is optimized by minimizing the following objective:

$$\mathcal{L}_{\mathrm{IOD}}^{(c)}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{train}}^{(c)}} \left[\ell_{\mathrm{JSD}}(\theta; x, y)\right]. \tag{14}$$

For trajectories in $\mathcal{A}^{(c)}$, distribution matching is performed under student-generated prefixes, whereas the same objective is evaluated under human-written reference prefixes for examples in $\mathcal{D}_H$. After optimization, the updated student is denoted by $\pi_S^{(c)}$ and is used to generate candidates in the subsequent cycle. The complete IOD procedure is summarized in Algorithm 1.

Algorithm 1 Iterative On-policy Distillation (IOD)

Require: Human-labeled data $\mathcal{D}_H$; fixed teacher π_T; initial student $\pi_S^{(0)}$; cycles C; candidates per input K; retention ratio q
Ensure: Distilled student $\pi_S^{(C)}$

$\mathcal{X}_H \leftarrow \{x_i : (x_i, y_i^H) \in \mathcal{D}_H\}$
$\mathcal{A}^{(0)} \leftarrow \varnothing$
for $c = 1, \ldots, C$ **do**
Grow: Student generation
  $\mathcal{P}^{(c)} \leftarrow \varnothing$
  for $x_i \in \mathcal{X}_H$ **do**
    for $k = 1, \ldots, K$ **do**
      Sample $\hat{y}_{i,k}^{(c)} \sim \pi_S^{(c-1)}(\cdot \mid x_i)$
      Add $(x_i, \hat{y}_{i,k}^{(c)})$ to $\mathcal{P}^{(c)}$
    end for
  end for
Evaluate/Filter: Trajectory selection
  $\widetilde{\mathcal{A}}^{(c)} \leftarrow \mathcal{A}^{(c-1)} \uplus \mathcal{P}^{(c)}$
  for all $(x, y) \in \widetilde{\mathcal{A}}^{(c)}$ **do**
    Obtain teacher score $s_T(x, y)$
  end for
  $\mathcal{A}^{(c)} \leftarrow \mathrm{TopQ}(\widetilde{\mathcal{A}}^{(c)}, s_T, q)$
Improve: Student update
  $\mathcal{D}_{\mathrm{train}}^{(c)} \leftarrow \mathcal{D}_H \uplus \mathcal{A}^{(c)}$
  Initialize π_θ from $\pi_S^{(c-1)}$
  Optimize $\mathcal{L}_{\mathrm{IOD}}^{(c)}(\theta)$ on $\mathcal{D}_{\mathrm{train}}^{(c)}$
  $\pi_S^{(c)} \leftarrow \pi_\theta$
end for
return $\pi_S^{(C)}$

4 Experiments

We evaluate IOD on XSum summarization (Narayan et al., 2018) and WMT14 English-to-German (En-De) translation (Bojar et al., 2014) using encoder-decoder and decoder-only models. We compare IOD with distillation baselines reproduced under matched experimental settings and analyze the effects of trajectory filtering, cross-cycle accumulation, and the student-update objective. We further examine retention-ratio sensitivity, with additional analyses of cycle-wise performance, teacher log-likelihood distributions, and computational cost provided in the appendix.

4.1 Experimental Setup

Datasets and evaluation metrics. We evaluate IOD on two conditional sequence-generation tasks. For abstractive summarization, we use XSum (Narayan et al., 2018), which contains news articles paired with single-sentence summaries. Summarization performance is evaluated using ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004). For machine translation, we use WMT14 English-to-German (En–De) (Bojar et al., 2014) and report BLEU (Papineni et al., 2002). We include WMT14 En–De to evaluate IOD in the translation setting considered in prior on-policy distillation research (Agarwal et al., 2024).

Teacher and student models. We consider both encoder–decoder and decoder-only architectures. For the encoder–decoder experiments, we use T5v1.1-XL as the teacher and T5v1.1-Small as the student (Raffel et al., 2020). For the decoder-only experiments, we use Qwen2.5-3B as the teacher and Qwen2.5-1.5B as the student (Yang et al., 2025). Both model configurations are evaluated on XSum and WMT14 En–De.

Table 1: Comparison of distillation methods on XSum summarization using encoder–decoder and decoder-only teacher–student configurations. ROUGE-1, ROUGE-2, and ROUGE-L are reported as absolute scores. All baseline methods are reproduced under the corresponding experimental setup, using the same initial teacher and student checkpoints as IOD.

Method	ROUGE-1	ROUGE-2	ROUGE-L
Encoder–decoder: T5v1.1-XL → T5v1.1-Small			
Student	38.625	16.097	30.964
KD	38.632	16.230	30.978
ImitKD	38.688	16.373	31.999
Distillm	38.903	16.798	33.446
GKD	38.829	16.875	32.974
IOD (ours)	**40.793**	**19.876**	**35.502**
Teacher	47.570	24.345	39.284
Decoder-only: Qwen2.5-3B → Qwen2.5-1.5B			
Student	40.483	17.109	32.343
Distillm	41.528	18.821	33.855
GKD	42.193	18.241	34.001
IOD (ours)	**42.452**	**19.093**	**34.119**
Teacher	42.730	19.101	34.296

Experimental protocol. We reproduce the distillation baselines within the corresponding IOD experimental setup, using matched task-specific teacher and student checkpoints, training data, and evaluation procedures. We report absolute task-specific scores rather than comparing teacher-relative performance gaps obtained from separate experimental pipelines. Unless otherwise specified, IOD uses three distillation cycles and a trajectory retention ratio of $q = 0.7$. To assess the variability of IOD, we conduct three random-seed runs for each combination of model architecture and dataset. This analysis is limited to IOD, and its results are reported separately in Appendix B.1. Further implementation details, including optimization, candidate generation, and evaluation configurations, are provided in Appendix A. We additionally report the hardware configurations and total wall-clock cost of IOD, including candidate generation, teacher scoring, and student optimization, in Appendix B.3.

4.2 Comparison with Distillation Baselines

We compare IOD with distillation baselines reproduced within the corresponding experimental setup for each task and model architecture. Unlike cross-study comparisons based on teacher-relative performance gaps, the present evaluation reports absolute task-specific scores obtained using the same task-specific teacher and student checkpoints, training data, and evaluation protocol within each configuration. We present the results for XSum summarization and WMT14 English-to-German (En–De) translation separately.

XSum summarization. Table 1 presents the XSum results for the encoder–decoder and decoder-only configurations. Under the matched experimental settings, IOD achieves higher ROUGE-1, ROUGE-2, and ROUGE-L scores than the reproduced baselines in both model architectures. The performance differences are more pronounced in the T5 configuration, whereas the Qwen2.5 results show a smaller improvement in ROUGE-L. These results demonstrate the applicability of IOD to both evaluated architectures, with the variability across random-seed runs examined separately in Appendix B.1.

WMT14 English-to-German translation. Table 2 presents the WMT14 En–De results for the encoder–decoder and decoder-only configurations. Under the matched experimental settings, IOD achieves higher BLEU scores than the reproduced baselines in both model architectures. These results complement the XSum findings by showing that the performance improvements observed with the complete IOD framework extend to the evaluated machine translation setting.

Table 2: Comparison of distillation methods on WMT14 English-to-German translation using encoder–decoder and decoder-only teacher–student configurations. Translation performance is measured using BLEU.

Method	BLEU
T5v1.1-XL → T5v1.1-Small	
Student	25.103
Distillm	25.684
GKD	25.584
IOD (ours)	**26.254**
Teacher	28.340
Qwen2.5-3B → Qwen2.5-1.5B	
Student	21.260
Distillm	22.891
GKD	22.870
IOD (ours)	**23.908**
teacher	27.703

4.3 Analysis of IOD Components

Trajectory filtering and cross-cycle accumulation. To evaluate the role of teacher-based filtering, we compare the full IOD configuration with a variant that retains all student-generated trajectories. We also consider a non-accumulating variant that applies the same teacher-based filtering criterion to the current-cycle candidate pool without reusing trajectories retained from earlier cycles.

As shown in Table 3, removing either filtering or accumulation leads to lower ROUGE-L scores than the full IOD configuration. These results indicate that both design choices affect performance in the evaluated setting. However, disabling accumulation also changes the size and composition of the student-generated training set. The comparison therefore measures the effect of the overall cross-cycle accumulation configuration, rather than isolating trajectory re-selection from differences in training-data volume. The cycle-wise performance of IOD across both model families and tasks is further reported in Appendix B.2.

Table 3: Ablation results on XSum using T5v1.1, measured by ROUGE-L. Full IOD uses teacher-based filtering, cross-cycle accumulation, and token-level Jensen–Shannon divergence (JSD). "Without filtering" retains all generated trajectories; "Without accumulation" filters only the current-cycle candidate pool. All student-update variants retain the full trajectory management procedure. REINFORCE uses sequence-level ROUGE-L rewards, and behavior cloning (BC) uses hard-label negative log-likelihood.

Configuration	ROUGE-L
Trajectory management	
IOD (full; JSD)	**35.50**
Without filtering	32.79
Without accumulation	31.01
Student-update objective	
JSD + REINFORCE	31.06
REINFORCE + BC	27.31
REINFORCE	30.96
BC	30.99

Student-update objectives. The default IOD configuration updates the student through token-level Jensen–Shannon divergence (JSD). To examine alternative supervision signals, we additionally evaluate REINFORCE (Williams, 1992), behavior cloning (BC), and their combinations with other objectives while retaining the corresponding trajectory-selection procedure.

For the REINFORCE variants, each selected student-generated summary receives a sequence-level reward based on its ROUGE-L score against the human-written reference. The reward weights the log-probability of the generated sequence under the trainable student, with a mini-batch baseline used to center the reward signal. BC instead optimizes the negative log-likelihood of the training sequences as hard targets. The mathematical definitions and implementation details of these alternative objectives are provided in Appendix A.5.

Table 3 shows that the JSD-only configuration achieves the highest ROUGE-L score among the evaluated student-update objectives. Adding REINFORCE to JSD or replacing JSD with the evaluated reward-based and hard-label objectives does not improve performance in this setting. We therefore retain token-level JSD as the default student-update objective of IOD. Extending GKD and DistiLLM to the same total number of student optimizer updates does not reproduce IOD's performance, indicating that the observed gains cannot be explained by additional training steps alone (Appendix A.4).

4.4 Sensitivity to the Retention Ratio

We investigate the sensitivity of IOD to the retention ratio q, which controls the fraction of student-generated trajectories retained after teacher-based filtering. Table 4 reports the results for different retention ratios using the T5v1.1 configuration on XSum. The $q = 100\%$ setting retains all generated trajectories, whereas the human-only control excludes student-generated trajectories from the training set. Among the evaluated settings, $q = 70\%$ achieves the highest ROUGE-L score, followed closely by $q = 50\%$. Retaining all generated trajectories or applying more restrictive filtering yields lower performance. Because the retention ratio changes both the number and composition of training trajectories, these results characterize sensitivity to the overall selection configuration rather than independently identifying the effect of teacher-score ranking. We use $q = 70\%$ as the default setting in the main experiments.

Table 4: Sensitivity to the retention ratio on XSum using the T5v1.1 configuration. The retention ratio q denotes the fraction of student-generated trajectories retained after teacher-based filtering. The $q = 0\%$ setting is a human-only control that uses the original human-labeled data without adding student-generated trajectories.

Setting	ROUGE-L
$q = 100\%$	32.79
$q = 70\%$	**35.50**
$q = 50\%$	35.13
$q = 10\%$	31.26
$q = 0\%$	30.94

4.5 Evolution of Teacher Log-Likelihood

We further examine how the fixed teacher evaluates student-generated outputs across successive IOD cycles. For each generated input–output pair (x, y), we compute the length-normalized conditional log-likelihood defined in Eq. 5. Since the teacher remains fixed throughout training, this score provides a consistent reference for examining changes in the outputs generated by successive student checkpoints. The teacher-score distributions for XSum and WMT14 En–De, together with their analysis, are provided in Appendix B.4.

5 Conclusion

We presented Iterative On-Policy Distillation (IOD), a multi-cycle framework that constructs an evolving training curriculum by jointly re-selecting previously retained and newly generated student trajectories under a fixed teacher criterion. IOD preserves the original human-labeled data and updates the student through token-level teacher–student distribution matching. Experiments on summarization and machine translation show that IOD achieves higher task-specific scores than reproduced distillation baselines across the evaluated encoder–decoder and decoder-only configurations. Ablation and teacher-score analyses further characterize the effects of IOD's components and the evolution of student-generated outputs. Future work will investigate compute-matched trajectory-management strategies to distinguish the effects of cross-cycle re-selection from changes in training-data volume.

AI USE STATEMENT

The authors used AI tools solely for language editing, grammar checking, and text polishing. All scientific content, analysis, and conclusions are the original work of the authors, who take full responsibility for the final manuscript.

ETHICS STATEMENT

This study investigates knowledge distillation for language generation using the XSum and WMT14 benchmarks. The proposed framework uses student-generated trajectories and a fixed teacher model for training-data selection and distribution matching. Although the framework is evaluated on summarization and translation tasks, distilled language models may inherit biases or generate inaccurate content from their training data and pretrained models. The reported results characterize performance under the evaluated datasets and experimental settings and should not be interpreted as establishing the reliability of generated outputs in other deployment contexts.

REPRODUCIBILITY STATEMENT

We provide the formulation and complete procedure of IOD in Section 3, including candidate generation, teacher-based trajectory selection, cross-cycle curriculum construction, and the student-update objective. Appendix A describes the datasets and evaluation protocols, model initialization and fine-tuning, candidate generation and teacher scoring, baseline reproduction, training budgets, and alternative student-update objectives. Appendix B reports additional experimental results, including random-seed variability, cycle-wise performance, computational cost, and the evolution of teacher log-likelihood. Appendix C further characterizes the mathematical properties of the trajectory selection and cross-cycle curriculum construction procedures.

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5be69a584901a26c521c2b51e40a4c20-Abstract-Conference.html.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/e995f98d56967d946471af29d7bf99f1-Abstract.html.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 12–58. Association for Computational Linguistics, 2014. doi: 10.3115/v1/W14-3302. URL https://aclanthology.org/W14-3302/.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. AlpaGasus: Training a better alpaca with fewer data. In *International Conference on Learning Representations*, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/9543942c237ded1b39b1fd37259ff88e-Abstract-Conference.html.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6621–6642. PMLR, 2024b. URL https://proceedings.mlr.press/v235/chen24j.html.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (ReST) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023. URL https://arxiv.org/abs/2308.08998.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

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. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895. PMLR, 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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31044–31062. PMLR, 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

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*, pp. 6121–6133. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.494. URL https://aclanthology.org/2020.emnlp-main.494/.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81. Association for Computational Linguistics, 2004. URL https://aclanthology.org/W04-1013/.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6091f2bb355e960600f62566ac0e2862-Abstract-Conference.html.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318. Association for Computational Linguistics, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319/.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL https://www.jmlr.org/papers/v21/20-074.html.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=lNAyUngGFK.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132. PMLR, 2024. URL https://proceedings.mlr.press/v235/xia24c.html.

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*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a2747a3844ca1e4667fbff3f558eb39b-Abstract-Conference.html.

An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 Technical Report, 2025. URL https://arxiv.org/abs/2412.15115.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488, 2022. URL https://arxiv.org/abs/2203.14465.

Supplementary Material

A Implementation Details

A.1 Datasets and Evaluation Protocols

Datasets. We evaluate IOD on two conditional sequence-generation tasks. For abstractive summarization, we use XSum, which contains news articles paired with single-sentence summaries. Summarization performance is evaluated using ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004). For machine translation, we use WMT14 English-to-German (En–De) and report BLEU (Papineni et al., 2002). We include WMT14 En–De to evaluate IOD in the translation setting considered in prior on-policy distillation research (Agarwal et al., 2024). Table 5 summarizes the dataset statistics and maximum input and output lengths used in our experiments.

Table 5: Dataset statistics and maximum input/output lengths used in our experiments. The maximum lengths are expressed in tokens.

Dataset	Task	Train	Validation	Test	Max. input/output
XSum	Summarization	204,045	11,332	11,334	1,024 / 64
WMT14 En–De	Translation	4,508,785	3,000	3,003	512 / 128

XSum contains BBC news articles paired with abstractive single-sentence summaries. Source documents are truncated to a maximum of 1,024 tokens, and target summaries are limited to 64 tokens. For WMT14, we load the `de-en` dataset configuration but perform translation in the English-to-German direction, using English sentences as inputs and German sentences as targets. The maximum source and target lengths are 512 and 128 tokens, respectively. Unless otherwise specified, we use the complete training split.

XSum evaluation. We evaluate summarization performance on the complete XSum test split of 11,334 examples. ROUGE-1, ROUGE-2, and ROUGE-L F1 scores are computed using the `rouge-score` implementation (Lin, 2004). For T5v1.1, we use beam search with four beams, while Qwen2.5 uses greedy decoding. Both configurations use deterministic generation without sampling, with a maximum of 64 newly generated tokens. Within each model family, identical decoding and evaluation settings are applied to all compared methods.

WMT14 evaluation. We evaluate translation performance on the complete WMT14 En–De test split of 3,003 examples. We report case-sensitive corpus BLEU scores (Papineni et al., 2002), computed using SacreBLEU (Post, 2018). For T5v1.1, we use beam search with four beams, while Qwen2.5 uses greedy decoding. Both configurations use deterministic generation without sampling, with a maximum of 128 newly generated tokens. Within each model family, identical decoding and evaluation settings are applied to all compared methods.

A.2 Model Initialization and Fine-tuning

Model configurations. We consider two teacher–student model families: LM-adapted T5v1.1 encoder–decoder models and Qwen2.5 decoder-only models. Table 6 summarizes their parameter counts and architectural configurations. For T5v1.1, the teacher is the LM-adapted XL model and the student is the LM-adapted Small model. For Qwen2.5, we use the 3B and 1.5B models as the teacher and student, respectively.

T5v1.1 supervised fine-tuning. We follow the supervised fine-tuning setup of prior on-policy distillation work (Agarwal et al., 2024) for the T5v1.1 teacher and student models. Both models are fine-tuned on the corresponding human-labeled training data using Adafactor.

Table 7 summarizes the task-specific training configurations. All T5v1.1 SFT runs use BF16 precision, gradient checkpointing, a maximum gradient norm of 1.0, and a linear learning-rate schedule. Weight decay is set to 0 for the teacher models and 0.01 for the student models.

Table 6: Teacher and student model configurations. Layer counts for T5v1.1 are reported as encoder/decoder layers.

Model	Parameters	Architecture	Layers
T5v1.1-Small	76.96M	Enc-dec	8/8
T5v1.1-XL	2.85B	Enc-dec	24/24
Qwen2.5-1.5B	1.544B	Decoder-only	28
Qwen2.5-3B	3.086B	Decoder-only	36

Table 7: SFT configurations for the T5v1.1 teacher and student models on XSum and WMT14 En-De. Training steps denote optimizer updates, and batch denotes the batch size. The learning rate increases linearly from zero to the peak LR during warmup.

Dataset	Model	Training Steps	Batch	Peak LR	Warmup schedule
XSum	T5-XL	8,000	128	10^{-4}	Linear ($0 \rightarrow$ LR), 2,000 steps warmup
XSum	T5-Small	100,000	128	10^{-3}	Linear ($0 \rightarrow$ LR) 2,000 steps warmup
WMT14	T5-XL	50,000	128	10^{-4}	Linear ($0 \rightarrow$ LR) 2,000 steps warmup
WMT14	T5-Small	250,000	128	10^{-3}	Linear ($0 \rightarrow$ LR) 10,000 steps warmup

Qwen2.5 supervised fine-tuning. We perform supervised fine-tuning of the Qwen2.5 teacher and student models on the corresponding human-labeled training data using AdamW. Table 8 summarizes the training configurations for XSum and WMT14 En-De.

Table 8: SFT configurations for the Qwen2.5 teacher and student models on XSum and WMT14 En–De. Training steps denote optimizer updates, and batch denotes the batch size. All configurations use linear warmup followed by cosine learning-rate decay.

Dataset	Model	Training Steps	Batch	Peak LR	LR schedule
XSum	Qwen2.5-1.5B	10,000	32	2×10^{-5}	Linear warmup (2,000 steps), Cos decay
XSum	Qwen2.5-3B	100,000	32	2×10^{-5}	Linear warmup (5,000 steps), Cos decay
WMT14	Qwen2.5-1.5B	50,000	512	2×10^{-5}	Linear warmup (2,000 steps), Cos decay
WMT14	Qwen2.5-3B	100,000	512	2×10^{-5}	Linear warmup (5,000 steps), Cos decay

All Qwen2.5 SFT runs use a peak learning rate of 2×10^{-5}, BF16 precision, and gradient checkpointing. The learning rate increases linearly from zero to the peak LR during warmup, followed by cosine decay. For WMT14 En–De, the batch size is 512, and the maximum prompt and target lengths are 512 and 128 tokens, respectively.

Distillation initialization. For each task and model family, we use the corresponding SFT teacher and student checkpoints to initialize distillation. All reproduced distillation baselines and IOD are initialized from the same task-specific SFT teacher and student checkpoints, ensuring consistent model initialization within each experimental configuration. The teacher remains fixed throughout the IOD cycles, while the student is updated through successive distillation steps.

For DistiLLM and GKD, we select the checkpoint with the highest validation ROUGE-L on XSum or corpus BLEU on WMT14.

For IOD, validation performance is monitored after each distillation cycle, and the final-cycle checkpoint is used for test evaluation. The test splits are not used for checkpoint selection.

A.3 Candidate Generation and Teacher Scoring

Candidate generation. At each distillation cycle c, the student checkpoint $\pi_S^{(c-1)}$ generates $K = 3$ candidate outputs for each input in the training set. We use stochastic decoding with sampling enabled, a nucleus-sampling threshold of $\texttt{top_p} = 0.9$, and a temperature of 1.0. The resulting candidate pool $\mathcal{P}^{(c)}$ contains input–output pairs generated by the student checkpoint at the

beginning of the corresponding cycle. Candidate generation uses training inputs only; test-set inputs and references are not used to construct the distillation data.

Teacher scoring. Each student-generated candidate is evaluated using the fixed, task-specific SFT teacher. We compute the length-normalized conditional log-likelihood defined in Eq. 5 over the generated output tokens, conditioned on the corresponding input and preceding output tokens. For decoder-only models, the task prompt provides the conditioning context and is excluded from the scored output tokens. The teacher remains fixed throughout distillation, and higher scores indicate greater conditional likelihood under the teacher.

Cross-cycle trajectory selection. At the first cycle, we retain the highest-scoring fraction q of the newly generated candidate pool. At each subsequent cycle, the previously retained set $\mathcal{A}^{(c-1)}$ is combined with the newly generated pool $\mathcal{P}^{(c)}$, and candidates are jointly ranked by their teacher scores. We retain the highest-scoring fraction of this combined pool, using $q = 0.7$ unless otherwise specified. Thus, previously retained trajectories remain eligible for selection but may be replaced by newly generated candidates in later cycles. The human-labeled dataset $\mathcal{D}_H$ is preserved independently of this filtering procedure, and the student is trained on $\mathcal{D}_H \cup \mathcal{A}^{(c)}$.

A.4 Baseline Reproduction and Training Budgets

Baseline reproduction. Within each task and model family, all reproduced distillation baselines and IOD use the same task-specific SFT teacher and student checkpoints, task datasets, and evaluation protocol. Each baseline retains its corresponding student-update procedure.

For the T5v1.1-based XSum experiments, KD and ImitKD are trained for 40,000 optimizer updates. GKD and DistiLLM are also evaluated at 40,000 updates and are further trained to examine their performance under larger training budgets. For WMT14 En–De, GKD and DistiLLM are evaluated at 150,000 updates and under extended training budgets.

Extended training-budget comparison. To examine whether additional optimizer updates alone account for the performance of IOD, we extend GKD and DistiLLM to the total student-update counts reached at the end of each IOD cycle. For IOD, this count is the sum of the optimizer updates performed during all completed Improve stages, excluding the preceding SFT stage. Table 9 reports the resulting task-specific scores alongside IOD performance at its cycle endpoints.

For XSum, both extended baselines attain their highest reported ROUGE-L scores at 40,000 updates, with lower scores at the subsequent evaluated checkpoints. On WMT14 En–De, both baselines attain their highest reported BLEU scores at 150,000 updates and do not exceed these scores at the later checkpoints. IOD achieves higher task-specific scores at each cycle endpoint than the corresponding extended baselines at the same total student-update count.

These results indicate that extending the optimization duration of the evaluated baselines does not reproduce IOD's performance under the tested configurations. Unlike uninterrupted baseline training, IOD periodically generates new student trajectories and re-selects them together with previously retained trajectories using a fixed teacher criterion, thereby updating the student-generated training curriculum.

This comparison aligns the total number of student optimizer updates at the IOD cycle endpoints, but does not establish identical processed-token counts, teacher-scoring costs, or total computational budgets. It therefore examines the effect of extending baseline training without isolating the individual contributions of trajectory selection, training-data volume, and the cycle-wise optimization schedule.

A.5 Alternative Student-update Objectives

We investigate alternative student-update objectives within the IOD framework using the T5v1.1-based XSum setting. These variants retain the candidate-generation, teacher-scoring, and trajectory-selection procedures of IOD while replacing or augmenting its default token-level JSD objective.

Table 9: Extended training-budget comparison using the T5v1.1 configuration on XSum and WMT14 En–De. Student training steps count optimizer updates performed after SFT initialization. For IOD, this count sums the updates across all completed Improve stages. GKD and DistiLLM are evaluated at the step counts corresponding to IOD cycle endpoints, with additional baseline checkpoints at 40,000 steps for XSum and 150,000 steps for WMT14 En–De. XSum results are reported using ROUGE-L F1, and WMT14 results using corpus BLEU. A dash indicates that no IOD cycle ends at the specified step count.

Dataset	IOD cycle	Student training steps (after SFT)	GKD	DistiLLM	IOD
XSum	Cycle 1	24,231	30.80	30.90	32.092
	–	40,000	32.97	33.45	–
	Cycle 2	60,960	31.20	31.30	33.724
	Cycle 3	106,437	30.70	31.10	35.502
WMT14 En–De	Cycle 1	134,230	25.32	25.54	25.780
	–	150,000	25.58	25.68	–
	Cycle 2	337,833	25.56	25.21	26.033
	Cycle 3	590,273	25.51	24.79	26.254

REINFORCE. We examine a sequence-level reward-based objective inspired by REINFORCE. For a student-generated summary y conditioned on a source document x, we define the reward as

$$R(x, y) = \text{ROUGE-L}(y, y^H), \tag{15}$$

where y^H denotes the corresponding human-written reference summary. This reward measures agreement between the generated summary and its reference, whereas teacher log-likelihood is used separately for trajectory selection.

For the REINFORCE update, we use only trajectories generated in the current cycle that are retained after teacher-based filtering. Let

$$\mathcal{R}^{(c)} = \left\{ z \in \mathcal{A}^{(c)} : \text{origin}(z) = c \right\}_{\text{multi}}, \tag{16}$$

where $\text{origin}(z)$ denotes the cycle in which a trajectory occurrence was generated. Thus, $\mathcal{R}^{(c)}$ contains the retained occurrences originating from $\mathcal{P}^{(c)}$, excluding trajectories carried over from earlier cycles.

The reward-weighted student-update objective is defined as follows:

$$\mathcal{L}_{\text{RL}}^{(c)}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{R}^{(c)}} \left[\big(R(x, y) - b\big) \sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid x, y_{<t}) \right], \tag{17}$$

where b denotes a reward baseline. When mini-batch reward centering is applied, b is the mean reward of the corresponding mini-batch. The reward and baseline are treated as constants when computing the student gradient.

This objective weights the log-probability of an observed sequence according to its sequence-level reward. In contrast, JSD provides token-level supervision by matching the teacher and student next-token distributions. Although the REINFORCE trajectories originate from the student checkpoint used for generation in the current cycle, they undergo teacher-based filtering and are subsequently used to update the student. Accordingly, this reward-weighted update is not generally an unbiased on-policy policy-gradient estimator for the trainable student.

Behavior cloning. We additionally evaluate behavior cloning (BC), which treats filtered student-generated summaries as hard supervision targets. The BC training data consist of the retained student-generated trajectories $\mathcal{A}^{(c)}$; human-written gold pairs (x, y^H) are not included in the BC training set. Its negative log-likelihood objective is as follows:

$$\mathcal{L}_{\text{BC}}^{(c)}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{A}^{(c)}} \left[\sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid x, y_{<t}) \right], \tag{18}$$

where y is the student-generated summary retained after teacher-based filtering, rather than the human-written reference y^H.

The original XSum documents provide the inputs for candidate generation, while their human-written summaries provide references for reward computation. BC increases the probability assigned to the retained student-generated target tokens without using sequence-level rewards or the teacher's probabilities over alternative tokens. In contrast, JSD uses the teacher's full next-token distribution to supervise the student on the selected training prefixes.

Objective variants. We evaluate REINFORCE and BC individually and in combination. We also examine a hybrid JSD–REINFORCE objective:

$$\mathcal{L}^{(c)}_{\mathrm{JSD+RL}}(\theta) = \lambda \mathcal{L}^{(c)}_{\mathrm{IOD}}(\theta) + (1-\lambda)\mathcal{L}^{(c)}_{\mathrm{RL}}(\theta), \tag{19}$$

where $\lambda = 0.5$.

For the REINFORCE–BC variant, we use

$$\mathcal{L}^{(c)}_{\mathrm{RL+BC}}(\theta) = \lambda_{\mathrm{RL}} \mathcal{L}^{(c)}_{\mathrm{RL}}(\theta) + \lambda_{\mathrm{BC}} \mathcal{L}^{(c)}_{\mathrm{BC}}(\theta), \tag{20}$$

with $\lambda_{\mathrm{RL}} = 1.0$ and $\lambda_{\mathrm{BC}} = 1.0$, corresponding to equal coefficients for the two loss terms.

These variants examine whether sequence-level reward feedback or hard-target supervision provides a useful alternative or complement to token-level teacher–student distribution matching under the evaluated trajectory-management procedure.

B Additional Experimental Results

B.1 Random-seed Analysis

We assess the variability of IOD across three independent random-seed runs for each model family and dataset. All runs use the same experimental configuration and are evaluated using the final-cycle student checkpoint ($C = 3$). Table 10 reports the mean and standard deviation of the resulting task-specific evaluation scores.

Table 10: Random-seed analysis of IOD at Cycle 3 across three independent runs. Results are reported as mean ± standard deviation. R-1, R-2, and R-L denote ROUGE-1, ROUGE-2, and ROUGE-L F1 scores, respectively.

Model family	Metric	Mean ± Std.
T5v1.1	XSum R-1	40.78 ± 0.14
	XSum R-2	19.87 ± 0.10
	XSum R-L	35.51 ± 0.12
	WMT14 BLEU	26.25 ± 0.13
Qwen2.5	XSum R-1	42.29 ± 0.29
	XSum R-2	19.06 ± 0.05
	XSum R-L	34.11 ± 0.12
	WMT14 BLEU	23.93 ± 0.07

B.2 Cycle-wise Performance Analysis

We examine how student performance changes across successive IOD cycles for both decoder-only and encoder–decoder configurations. Cycle 0 denotes the SFT-initialized student before distillation, while Cycles 1–3 denote the student checkpoints obtained after successive IOD updates. All checkpoints within each model family are evaluated using the same task-specific evaluation protocol.

Qwen2.5 decoder-only models. Table 11 reports the cycle-wise results for Qwen2.5 on XSum and WMT14 En–De. Performance improves from the SFT-initialized student to Cycle 3 across all reported metrics. The results show that the successive IOD updates are accompanied by improvements in both summarization and translation performance under the evaluated decoder-only configuration.

Table 11: Cycle-wise performance of the Qwen2.5 decoder-only configuration on XSum summarization and WMT14 English-to-German translation. Cycle 0 denotes the SFT-initialized student, and Cycles 1–3 denote successive IOD updates. R-1, R-2, and R-L denote ROUGE-1, ROUGE-2, and ROUGE-L F1 scores, respectively.

Model / Cycle	XSum R-1	XSum R-2	XSum R-L	WMT14 En–De BLEU
Student (Cycle 0)	40.483	17.109	32.343	21.260
IOD (Cycle 1)	42.034	18.503	33.720	23.508
IOD (Cycle 2)	42.271	18.715	33.895	23.692
IOD (Cycle 3)	**42.452**	**19.093**	**34.119**	**23.908**
Teacher	42.730	19.101	34.296	27.703

Table 12: Cycle-wise performance of the T5v1.1 encoder–decoder configuration on XSum summarization and WMT14 English-to-German translation. Cycle 0 denotes the SFT-initialized student, and Cycles 1–3 denote successive IOD updates. R-1, R-2, and R-L denote ROUGE-1, ROUGE-2, and ROUGE-L F1 scores, respectively.

Model / Cycle	XSum R-1	XSum R-2	XSum R-L	WMT14 En–De BLEU
Student (Cycle 0)	38.625	16.097	30.964	25.103
IOD (Cycle 1)	39.840	17.238	32.092	25.780
IOD (Cycle 2)	40.624	18.371	33.724	26.033
IOD (Cycle 3)	**40.793**	**19.876**	**35.502**	**26.254**
Teacher	47.570	24.345	39.284	28.340

T5v1.1 encoder–decoder models. Table 12 presents the corresponding cycle-wise evaluation for the T5v1.1 encoder–decoder configuration. As in the Qwen2.5 analysis, we compare the SFT-initialized student with the checkpoints obtained after each IOD cycle using ROUGE on XSum and corpus BLEU on WMT14 En–De.

We use three distillation cycles as the default experimental setting; the cycle-wise results characterize performance within this evaluated horizon rather than establishing an optimal stopping point.

B.3 Computational Cost

Table 13 reports the hardware configurations and total wall-clock time of the main IOD experiments. The reported runtime includes candidate generation, teacher scoring, and student optimization across distillation cycles. Since the T5v1.1 and Qwen2.5 experiments use different GPU configurations, their wall-clock times are reported as implementation-level costs rather than directly comparable measures of computational efficiency.

Table 13: Hardware and total wall-clock cost of the main IOD experiments. Runtime includes candidate generation, teacher scoring, and student optimization.

Experiment	GPU	# GPUs	Total Time
XSum, T5	A100 (80GB)	1	147 h 27 min
WMT14, T5	A100 (80GB)	1	153 h 31 min
XSum, Qwen2.5	H200 (140GB)	4	29 h 56 min
WMT14, Qwen2.5	H200 (140GB)	4	38 h 23 min

B.4 Evolution of Teacher Log-Likelihood

We examine how the fixed teacher evaluates outputs generated by successive IOD student checkpoints on held-out test inputs using the T5v1.1 encoder–decoder configuration. At each cycle, the student independently generates summaries or translations for the corresponding test inputs. For each generated input–output pair (x, y), we compute the length-normalized conditional log-

likelihood $s_T(x, y)$ defined in Eq. 5. The teacher remains fixed across cycles, and human references are not used in this scoring procedure.

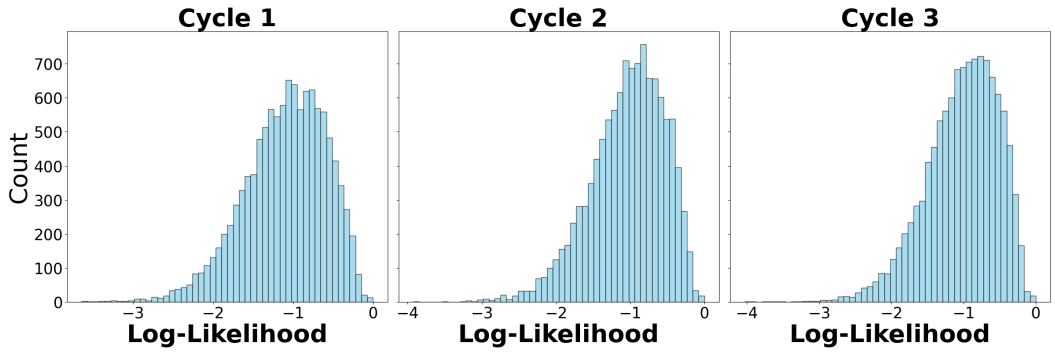

Figure 2: Distributions of length-normalized teacher log-likelihood for summaries generated by successive T5v1.1-Small student checkpoints on the XSum test set. Each panel corresponds to one IOD cycle. Scores closer to zero indicate higher conditional likelihood under the fixed T5v1.1-XL teacher.

Figure 2 shows a gradual shift toward higher teacher log-likelihood values across the evaluated cycles. This pattern indicates that the fixed teacher assigns relatively higher likelihood to the test-set summaries generated by later student checkpoints.

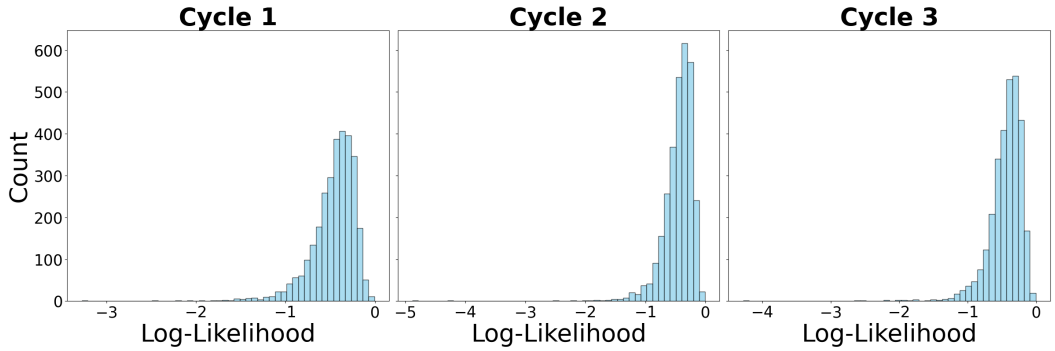

Figure 3: Distributions of length-normalized teacher log-likelihood for English-to-German translations generated by successive T5v1.1-Small student checkpoints on the WMT14 En–De test set. Each panel corresponds to one IOD cycle. Scores closer to zero indicate higher conditional likelihood under the fixed T5v1.1-XL teacher.

Figure 3 presents the corresponding analysis for WMT14 En–De. Compared with Cycle 1, outputs generated in later cycles are more concentrated in higher-likelihood regions, indicating a similar change in teacher compatibility for the evaluated translation task.

Importantly, these histograms are computed from newly generated test-set outputs rather than the candidate pools or retained trajectories used during training. Consequently, the observed distributions do not merely reflect the direct selection of high-scoring training samples. Instead, they characterize how the outputs of successive student checkpoints are evaluated under a common, fixed teacher on test inputs. This analysis complements the task-specific evaluation results, but higher teacher likelihood does not necessarily imply higher ROUGE or BLEU, nor does the observed shift independently isolate the effects of trajectory filtering or cross-cycle re-selection.

C MATHEMATICAL PROPERTIES OF IOD

Teacher-based trajectory selection. For a fixed candidate pool and retention size, TopQ selects the trajectories with the highest teacher scores and maximizes their total score among all subsets of the same size. This property concerns the specified teacher-score criterion and does not guarantee lower distillation loss or higher downstream task quality.

Cross-cycle curriculum construction. At each cycle, IOD jointly re-selects previously retained trajectories and newly generated candidates. Let $n_c = |\mathcal{A}^{(c)}|$ and $M_c = |\mathcal{P}^{(c)}|$ denote the realized number of retained trajectories and the number of newly generated candidates, respectively. Under the multiset concatenation defined in the main text, the size of the combined candidate pool is

$$\widetilde{n}_c = n_{c-1} + M_c, \qquad n_0 = 0, \tag{21}$$

where $\widetilde{n}_c = |\widetilde{\mathcal{A}}^{(c)}|$.

Let τ_c denote the empirical $(1-q)$-quantile of the teacher scores in the combined candidate pool:

$$\tau_c = Q_{1-q}\left(\left\{s_T(z) : z \in \widetilde{\mathcal{A}}^{(c)}\right\}_{\text{multi}}\right), \tag{22}$$

where Q_{1-q} is computed using linear interpolation.

The filtering procedure retains all trajectory occurrences whose teacher scores are at or above the threshold, including those tied at the cutoff. The realized retained-set size is therefore

$$n_c = \sum_{z \in \widetilde{\mathcal{A}}^{(c)}} \mathbf{1}\left[s_T(z) \geq \tau_c\right], \tag{23}$$

where the sum counts individual trajectory occurrences in the multiset.

For planning the student optimization schedule, we use the nominal top-q recurrence

$$n_c^{\text{plan}} = \left\lceil q\left(n_{c-1}^{\text{plan}} + M_c\right)\right\rceil, \qquad n_0^{\text{plan}} = 0. \tag{24}$$

The realized size n_c may differ from n_c^{plan} due to linear interpolation in quantile estimation and the retention of all trajectories tied at the threshold. Such differences may also propagate across cycles because the realized retained collection is used to construct the next candidate pool.

When the number of newly generated candidates is fixed across cycles, the nominal retention count n_c^{plan} is nondecreasing under the stated recurrence. This property does not necessarily hold for the realized retained-set size n_c under quantile-based filtering.

Previously selected trajectories may be replaced by newly generated candidates, so the retained collections are not necessarily nested. Moreover, cross-cycle accumulation changes both the candidate composition and the amount of student-generated training data. Under a fixed number of training epochs per cycle, changes in training-set size can also change the number of optimizer updates.

Consequently, an accumulation ablation that changes these factors jointly does not isolate the effect of cross-cycle re-selection alone.

Finally, the selection rule guarantees neither monotonic improvement in the mean teacher score of the retained trajectories nor monotonic improvement in downstream task performance across cycles.

Distribution matching on selected prefixes. Teacher-based selection ranks completed student-generated trajectories, whereas the JSD objective matches teacher and student next-token distributions on prefixes in the selected training data. At cycle c, this objective is applied to human-reference prefixes from $\mathcal{D}_H$ and selected student-generated prefixes from $\mathcal{A}^{(c)}$.

New candidates originate from the student checkpoint used for generation in the current cycle. Previously retained trajectories, however, may originate from earlier student checkpoints and are not necessarily on-policy with respect to the current student. Thus, low divergence on selected training prefixes does not by itself guarantee low divergence on unseen inference-time prefixes or improved downstream task performance.