AdaSwitch: Balancing Exploration and Guidance in Knowledge Distillation via Adaptive Switching

Anonymous ACL submission

Abstract

Small language models (SLMs) are crucial for applications with strict latency and computational constraints, yet achieving high performance remains challenging. Knowledge distillation (KD) can transfer capabilities from large teacher models, but existing methods face a dilemma: off-policy distillation provides high-quality supervision but suffers from exposure bias (training–inference mismatch), while on-policy approaches ensure consistency but are limited by the low quality of student-generated outputs. To address these issues, we propose AdaSwitch, a novel approach that dynamically combines on-policy and off-policy generation via an adaptive switching mechanism. AdaSwitch allows the student to explore its predictions within its capability and selectively integrates teacher guidance only when divergence exceeds a context-aware threshold. This paradigm preserves generation consistency while ensuring high-quality supervision. Experiments on three datasets demonstrate that AdaSwitch consistently improves accuracy and reasoning capability with moderate overhead. Our code is available at: `https://anonymous.4open.science/r/AdaSwitch-76F9`

1 Introduction

With the rapid development of large language models (LLMs), scaling laws suggest that increasing parameters generally leads to better performance. However, in latency-sensitive scenarios such as real-time search and recommendation systems, deploying massive LLMs is often computationally prohibitive (Li et al., 2025; Deng et al., 2025). Therefore, developing Small Language Models (SLMs) that retain strong performance while minimizing inference costs is of vital importance.

Obtaining high-quality SLMs is often achieved by transferring knowledge from large teacher models via Knowledge Distillation (KD) (Hinton et al., 2015; Xu et al., 2024; Wang et al., 2025; Sanh et al., 2019). Current KD paradigms generally fall into two categories depending on the source of training data (Xu et al., 2024): Off-policy methods (e.g., MiniLLM (Gu et al., 2024), SeqKD (Kim and Rush, 2016)) utilize teacher-generated sequences or ground-truth data, providing rich supervision signals but suffering from training-inference mismatch, as the student never learns to recover from its own errors. Conversely, on-policy methods (e.g., GKD (Agarwal et al.)) train on student-generated sequences, aligning training with inference but often collapsing due to the low quality of the student's initial outputs, especially in the early stages of distillation. Effectively combining the strengths of both paradigms—high-quality supervision and distribution consistency—remains a key challenge.

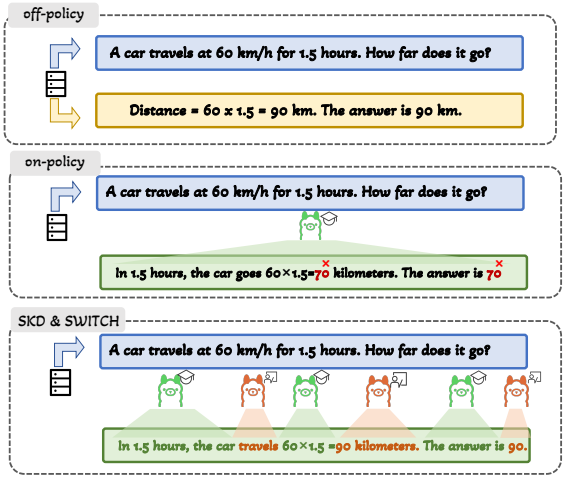

Figure 1: An example illustrating the sequence generation process for off-policy, on-policy, SKD, and SWITCH methods. denotes the teacher, and represents the student.

Recent attempts, such as SKD (Xu et al.) and SWITCH (Koo et al., 2025), propose mixing student and teacher tokens within a single sequence. However, these methods typically employ frequent, bidirectional switching strategies, where control is

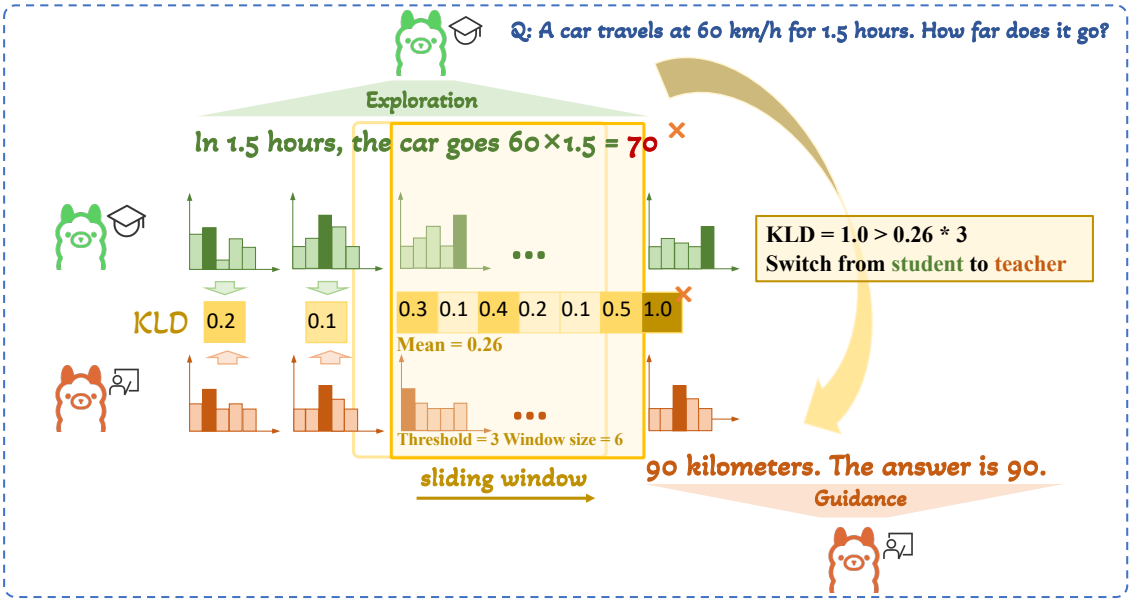

Figure 2: Overview of the proposed AdaSwitch approach. When the divergence between the student and teacher logits for the next token exceeds $\tau = K \cdot \bar{d}_i$, AdaSwitch switches to the teacher to generate the remaining tokens. denotes the teacher, and represents the student.

returned to the student immediately after a teacher intervention. We argue that such fragmentation disrupts the semantic coherence of the generation process. When a model is forced to constantly toggle between the student's and the teacher's latent states, it may generate disjointed sequences (or "chimeras") that satisfy neither distribution perfectly. Furthermore, existing static or rigid thresholding mechanisms fail to adapt to the varying difficulty of different samples, limiting their effectiveness on complex reasoning tasks.

To address these limitations, we propose **AdaSwitch**, a context-aware sequence generation framework that dynamically alternates between on-policy exploration and off-policy guidance at the token level. AdaSwitch adopts an adaptive one-time switching mechanism: the student initially generates tokens autonomously (Exploration Stage) to maintain training-inference consistency. Once the student's divergence from the teacher exceeds a dynamic threshold derived from recent generation history, the teacher takes over to generate the remaining sequence (Guidance Stage), ensuring high-quality outputs. This design prevents logic fragmentation caused by frequent switching and ensures that the student receives a high-quality correction exactly when its capability limit is reached.

We evaluate AdaSwitch on three datasets using three distinct teacher–student model pairs. Experimental results reveal that AdaSwitch consistently improves performance in most scenarios, highlighting its robustness and versatility across tasks. In terms of efficiency, AdaSwitch maintains an acceptable computational overhead, averaging only 1.3× that of the pure on-policy method, while achieving a 10% reduction compared with SKD and SWITCH. This modest overhead increase ensures that AdaSwitch remains highly practical for real-world applications.

2 Methodology

2.1 Problem Setup

We study KD for LLMs by considering a teacher model M_t and a student model M_s with different capacities. Our goal is to train M_s to mimic M_t's behavior on a specific task T. The student model M_s is parameterized by learnable weights θ_s, while the teacher model M_t has fixed parameters θ_t.

For task T, we assume access to a collection of input prompts x and associated target sequences y, either from ground-truth data or model-generated outputs. To quantify the difference between the teacher's and student's token-level predictions, we define a divergence function $D(\cdot \,\|\, \cdot)$, such as Kullback–Leibler (KL) divergence:

$$d_i(M_t, M_s) = D\big(M_t(\cdot \mid y_{<i}, x) \,\|\, M_s(\cdot \mid y_{<i}, x)\big), \tag{1}$$

Algorithm 1 AdaSwitch Knowledge Distillation for LLMs

Input: Student LLM M_s, Teacher LLM M_t, Prompt dataset $\{x_n\}_{n=1}^{N}$, Decoding length α, Sliding window length L, Divergence threshold multiplier K, Divergence metric D

```
for n = 1 to N do
    Initialize empty list of recent divergences D_window
    use_teacher ← False
    for i = 1 to α do
        if use_teacher then
            Sample y_i ~ M_t(· | y_<i, x_n)                          ▷ teacher continues generating
        else
            Sample y_i ~ M_s(· | y_<i, x_n)                          ▷ student generates token
            Compute divergence d_i = D(M_t(· | y_<i, x_n) ‖ M_s(· | y_<i, x_n))
            if i > L then
                Compute sliding window average d̄_{i-1} = (1/L) Σ_{j=i-L}^{i-1} d_j
                if d_i > K · d̄_{i-1} then
                    use_teacher ← True                               ▷ switch to teacher for remaining tokens
                    Sample y_i ~ M_t(· | y_<i, x_n)
                end if
            end if
            Append d_i to D_window
        end if
    end for
    Apply gradient descent to minimize D(M_t‖M_s)(y | x_n) over generated sequence
end for
```

where i indexes the decoding step. The overall sequence-level divergence is then:

$$D(M_t \| M_s)(y|x) = \frac{1}{L_y} \sum_{i=1}^{L_y} d_i(x, y), \qquad (2)$$

with L_y the length of y. The learning objective is to minimize $D(M_t \| M_s)(y|x)$, encouraging M_s to closely approximate M_t.

2.2 Target Sequence Selection

The choice of target sequence y is critical in KD. Off-policy methods use fixed targets from datasets, while on-policy methods generate y from the student model itself:

$$y \sim M_s(\cdot \mid x; \theta_s). \qquad (3)$$

However, off-policy methods suffer from a mismatch between training and inference in the student model, whereas on-policy methods are limited by the low quality of student-generated sequences, leading to suboptimal performance.

2.3 AdaSwitch

To overcome the limitations of off-policy and on-policy approaches and fully unlock the potential of knowledge distillation, we propose a novel sequence generation approach known as **AdaSwitch** as shown in Figure 2.

Adaswitch is designed to enhance the training of student models during sequence generation through dynamic guidance based on the divergence between the predictions made by the student and the teacher models. The primary objective of AdaSwitch is to achieve a delicate balance between exploration and guidance: while the student model is permitted to explore early predictions, it receives corrective interventions when its outputs diverge significantly from those of the teacher, thereby ensuring high-quality supervision throughout the training process.

At each generation step i, we compute the divergence $d_i(M_t, M_s)$ between the next-token predictions from the teacher and the student, as defined in Equation 1. This divergence serves as a quantitative measure of the discrepancy between the two models' predictions on the current token, allowing us to evaluate the alignment of the student with the teacher's knowledge.

To effectively assess whether the current predictions of the student model diverge from those of the teacher model, we introduce a metric that adapts to sequences characterized by varying distributions. Specifically, we maintain a sliding window of length L over the most recent divergences, computing the moving average as follows:

$$\bar{d}_i = \frac{1}{L} \sum_{j=i-L+1}^{i} d_j, \quad \text{for } i \geq L. \qquad (4)$$

This moving average, $\bar{d}_i$, provides a context-sensitive measure of divergence, allowing the model to adjust its behavior according to recent

	Llama-3.1-3b to Llama-3.1-1b			Qwen-2.5-3b to Qwen-2.5-0.5b			Gemma-7b to Gemma-2b		
	SUMM	GSM	GSM_Plus	SUMM	GSM	GSM_Plus	SUMM	GSM	GSM_Plus
Teacher	37.5	59.6	38.6	37.4	65.4	45.4	57.0	36.9	33.4
Student	13.5	0.0	0.0	12.1	0.0	0.0	11.6	0.0	0.0
SFT	32.3	29.0	12.0	30.3	31.0	15.5	31.8	18.9	11.7
SeqKD	32.1	28.2	11.8	30.1	30.4	14.4	31.0	21.8	10.5
Off-policy	34.6	30.6	13.8	32.4	33.5	17.9	35.1	22.6	10.7
On-policy	34.3	32.7	14.3	32.3	34.1	20.2	34.1	26.2	13.8
ImitKD	35.1	30.6	15.6	32.5	34.6	21.3	34.9	25.3	13.5
SWITCH	33.7	33.2	16.0	31.8	35.2	20.8	35.1	24.8	13.8
SKD	34.9	32.4	16.9	32.3	33.0	20.4	35.0	26.0	13.9
AdaSwitch	**35.6**	**35.0**	**17.3**	**32.6**	**36.3**	**21.6**	**35.6**	**26.8**	**14.2**

Table 1: Performance comparison of different Knowledge Distillation methods for Llama and Qwen on three tasks. Performance is evaluated using Rouge-L on the SUMM summarization dataset and accuracy on the GSM and GSM-Plus reasoning benchmarks.

prediction patterns. Based on this, we define a threshold τ when generating y_i as follows:

$$\tau = K \cdot \bar{d}_{i-1}, \tag{5}$$

where K is a hyperparameter that scales the threshold relative to the moving average of divergences. The student model continues generating tokens as long as its divergence from the teacher model remains below this adaptive threshold τ. If the divergence exceeds τ, the teacher continues to generate:

$$y_i \sim \begin{cases} M_s(\cdot \mid y_{<i}, x; \theta_s), & d_i(M_t, M_s) \leq \tau, \\ M_t(\cdot \mid y_{<i}, x; \theta_t), & d_i(M_t, M_s) > \tau. \end{cases} \tag{6}$$

Once a switch to the teacher model occurs, all subsequent tokens are generated exclusively by the teacher model, eliminating the need for further divergence checks. The complete algorithm of AdaSwitch is presented in Algorithm 1.

By integrating adaptive sliding-window thresholds and selective teacher intervention, AdaSwitch not only ensures consistency between training and inference but also maintains high-quality supervision for the student model. This approach effectively harnesses the strengths of both exploration and guidance, leading to improved model performance and robustness in diverse sequence generation tasks.

3 Experiments

3.1 Student and Teacher Models

We conducted experiments using three state-of-the-art LLM families: Qwen 2.5 (Team, 2024), Llama 3.1 (Grattafiori et al., 2024) and Gemma (Team et al., 2024). For the teacher models, we employed the SFTed Qwen-2.5-3B, Llama-3.1-3B and Gemma-7B, while the student models consisted of Qwen-2.5-0.5B, Llama-3.1-1B and Gemma-2B.

3.2 Datasets and Experimental Setup

To evaluate the effectiveness of our proposed AdaSwitch, we conduct experiments on two distinct types of datasets: DialogSum (Chen et al., 2021) for dialogue summarization and GSM (Cobbe et al., 2021) for arithmetic reasoning. For arithmetic reasoning, we use one additional dataset GSM-Plus (Li et al., 2024) to further test the generalizability and robustness of our approach. Each task follows a two-stage procedure: we first fine-tune the teacher model on the task, under the assumption that it serves as a domain expert, and then perform knowledge distillation by randomly sampling approximately 1K examples from the task, following common practices in prior KD work. Detailed configurations for each dataset are provided in the subsequent sections. For reproducibility, all implementations adopt greedy decoding during evaluation. Detailed descriptions of the datasets, including metrics and data splits, are provided in Appendix A.

3.3 Hyperparameters

For all SFT experiments, we used a learning rate of 1e-5, a warmup ratio of 0.1, and a dropout rate of 0.1, training each model for three epochs and selecting the checkpoint with the lowest validation loss. For KD, input/output lengths were set to 1024, 256 for dialogue summarization and 128, 512 for arithmetic reasoning. Student models used a temperature of 0.5 and top-p of 0.5, while teachers used

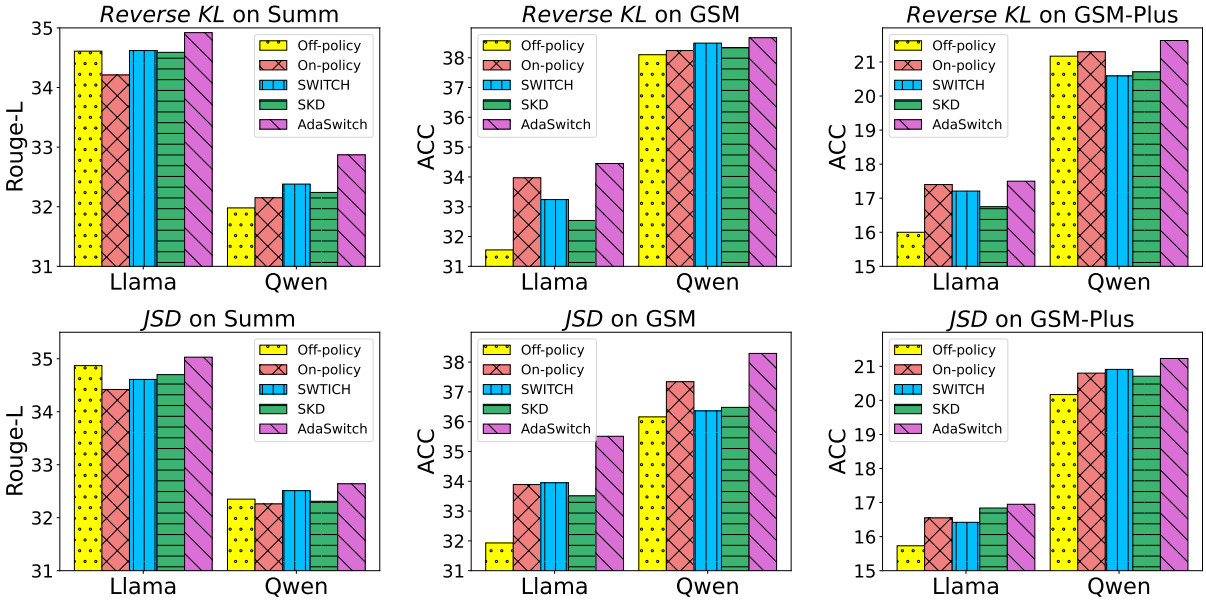

Figure 3: Further comparison of performance under different distance metrics on three tasks.

a temperature of 0.2 and top-p of 0.5, following prior work (Xu et al.). All models were trained for three epochs with a learning rate of 1e-5. For fair comparison, we set $K_{skd} = 0.5$ for SKD, and fixed the sliding window length $L = 10$ and threshold $K = 3$. The effects of L and K are analyzed later.

3.4 Baselines

Supervised FT directly minimizes the negative log-likelihood of the student's predictions on the target sequence, which is the ground-truth.

SeqKD (Kim and Rush, 2016) performs Supervised FT on teacher-generated target sequences.

SupervisedKD (Sanh et al., 2019) trains a student model to mimic the token-level probability distribution of a teacher model over a fixed ground-truth sequence of each (x, y) pair by minimizing the distance between the student's and teacher's predictions. Since this method delivers the best performance among all off-policy approaches, we refer to it simply as **off-policy** in the following.

GKD (Agarwal et al.) applied on-policy KD to address the training-inference mismatch in student models by sampling target tokens directly from the student's own output. We refer to this method as **on-policy** in the remainder of this paper.

ImitKD (Lin et al., 2020) randomly sample target sequences from either the ground truth dataset or the student model, which is a sequence-level mixing strategy of on-policy and off-policy.

SWITCH (Koo et al., 2025) employs token-level switching between the student and teacher based on a fixed threshold decay during sequence generation.

SKD (Xu et al.) employs a speculative decoding method to improve the quality of the generated sequences by using the teacher model to supervise the sequence generation.

3.5 Overall Performance

As shown in Table 1, on-policy and off-policy approaches exhibit complementary strengths. On-policy distillation outperforms on GSM and GSM-Plus, whereas off-policy achieves superior results on SUMM. These observations suggest that integrating both methods can be advantageous, allowing the student model to benefit from the strengths of each approach.

Mixed methods such as ImitKD, SWITCH and SKD attempt to integrate both paradigms and yield localized improvements. For instance, ImitKD performs well on SUMM, and SKD shows gains on GSM-Plus. However, these improvements are marginal and inconsistent across tasks and model pairs, reflecting the limited performance improvement of their designs.

In contrast, AdaSwitch achieves state-of-the-art results across all benchmarks, demonstrating both robustness and effectiveness. On GSM, AdaSwitch surpasses the second-best method by 7.2% and 11.8% when applied to Llama and Qwen, respectively. These results highlight its ability to integrate the strengths of on-policy and off-policy learning at the token level, leading to significant and consistent improvements.

For a more comprehensive comparison, we evaluate performance using JSD and reversed KLD

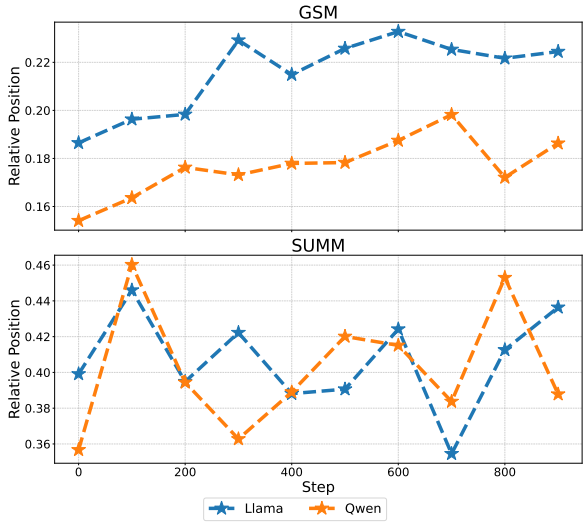

Figure 4: Analysis of the relative switching position of the distillation process

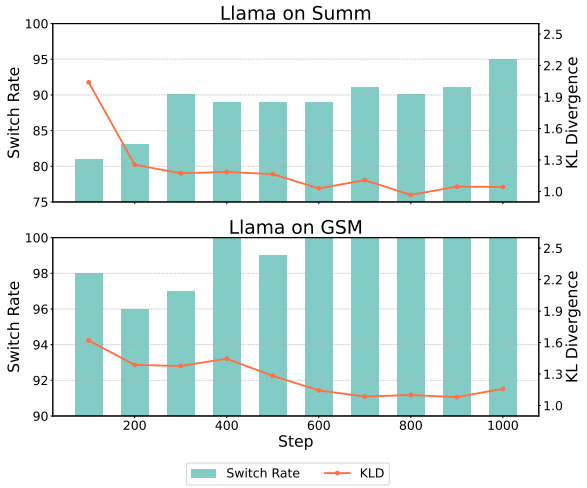

Figure 5: Analysis of the switch rate and the KLD at the switch token between the student and teacher models throughout the distillation process. Statistics were collected every 100 steps for the first 1000 steps.

(Gu et al., 2024) as distance metrics, as shown in Figure 3. The results indicate that, while existing mixed KD methods occasionally provide improvements, their performance is inconsistent. By comparison, although results vary slightly across different metrics, AdaSwitch consistently performs well and is compatible with different commonly used distance measures, highlighting the robustness and flexibility of AdaSwitch in diverse scenarios.

3.6 Analysis of Switching Dynamics during Distillation

To better understand the distillation process of AdaSwitch, we analyze the switch rate, defined as the proportion of sequences that trigger a switch from exploration to guidance, along with the average KLD between the student and teacher at the switching points. Statistics were collected every 100 steps, and the results are shown in Figure 5.

First, we observe a significant decrease in KLD during distillation across all tasks. This trend indicates that the student model gradually converges toward the teacher, consistent with intuitive expectations. Moreover, the KLD in the SUMM task is consistently lower than that in GSM, aligning with the performance difference between the two tasks and indicating that the SUMM task presents a lower reasoning complexity compared to GSM.

Switch rates also reveal task-dependent differences. On GSM, switch rates remain above 95% throughout, whereas SUMM exhibits substantially lower rates. This suggests that more challenging tasks demand stronger reliance on teacher guidance, while easier tasks allow greater exploration. AdaSwitch thus adaptively balances exploration and guidance according to task difficulty.

Further, we observe that the switch rate increases steadily during distillation, most notably in SUMM, while remaining near 100% in GSM. This trend reflects a gradual decline in exploration and a rise in guidance, indicating that on-policy distillation contributes less over time. Although this may seem counterintuitive, since weaker students would normally require more guidance, the pattern is explained by KLD: early in training, high divergence suppresses switching, but as the student aligns with the teacher, switching becomes more frequent. In other words, AdaSwitch begins in an exploration-dominated on-policy mode and gradually shifts to off-policy guidance as the student improves. This adaptive behavior allows AdaSwitch to dynamically balance exploration and guidance according to the student's capability, ensuring optimal distillation across varying tasks.

Moreover, we analyze the relative switching position of AdaSwitch during distillation. This also reflects the proportion of tokens generated by the student model. As shown in Figure 4, in GSM the switching position exhibits a clear upward trend, indicating that the teacher intervenes later in the generation process. This suggests that the student model is gradually approaching the teacher's capability. In contrast, for the SUMM task we do not observe a noticeable increase, which we attribute to the smaller performance gap between the student and teacher models in this setting.

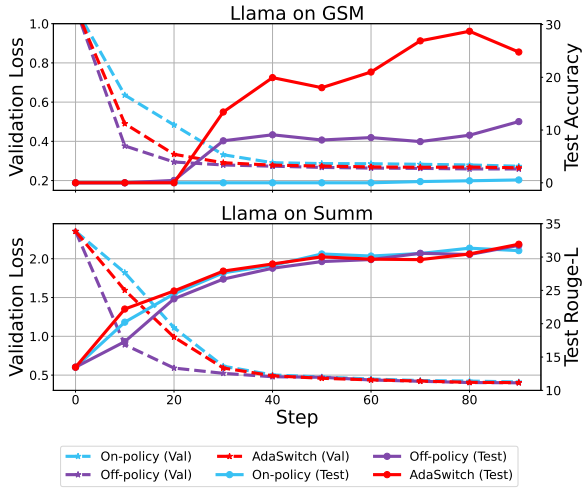

Figure 6: Performance on the validation and test sets during the early stages of distillation.

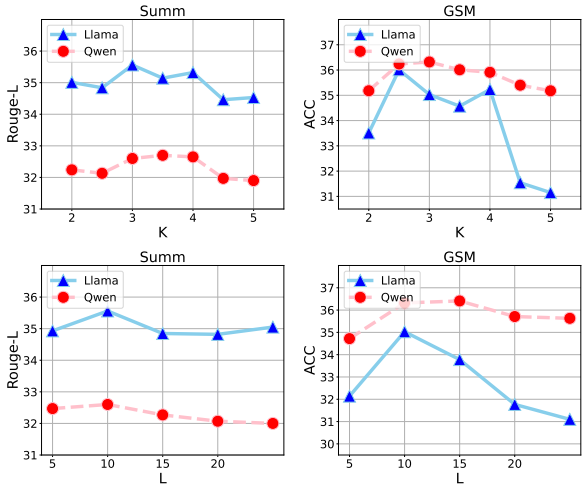

Figure 7: Parameter analysis on K and L.

3.7 Early-Stage Distillation Analysis

The challenge of low-quality outputs in on-policy KD is particularly acute during the early stages of the distillation process. To investigate whether AdaSwitch addresses this issue, we present the validation loss and test performance trends for on-policy, off-policy, and AdaSwitch methods over the first 100 steps in Figure 6.

Regarding the validation loss trends, the result indicates that AdaSwitch consistently achieves values intermediate between those of the on-policy and off-policy methods, providing evidence of an effective integration. Since validation loss is computed using the teacher model's outputs as ground truth, the off-policy method, which directly optimizes against these outputs, attains the lowest validation loss.

Further experimental results of the performance on the test set reveal that AdaSwitch demonstrates the most rapid improvement rate, particularly on the GSM dataset. Notably, at step 30, the accuracy of AdaSwitch surpasses 10%, whereas the off-policy and on-policy methods achieve only 7.96% and 0%, respectively. These results suggest that AdaSwitch, through effective token-level mixing, simultaneously ensures the quality of generated sequences and reduces the discrepancy between training and inference sequences during the early stages of distillation. The full experimental results are presented in Appendix B.

3.8 Parameter Analysis

In the proposed AdaSwitch framework, two hyperparameters, K and L, play pivotal roles: K regulates the threshold for switching between exploration and guidance modes, while L determines the window size for this switching mechanism.

Regarding K, empirical results indicate that optimal performance is achieved when K lies between 3 and 4. Increasing K tends to elevate the proportion of exploration relative to guidance, whereas decreasing K has the opposite effect. When comparing performance across datasets, we observe that the impact of varying K is relatively minor and more stable on the SUMM dataset. In contrast, on the GSM dataset, particularly for the Llama model, performance is significantly more sensitive to changes in K.

For the parameter L, the best results are generally obtained around a value of 10. Larger values of L increase the minimum duration of exploration (i.e., at least L tokens), while simultaneously reducing the sensitivity of the switching mechanism to fluctuations in the KLD. Conversely, smaller L values increase sensitivity to KLD changes. Consistent with observations for K, the effects of varying L are more stable and less pronounced on the SUMM dataset, whereas on the GSM dataset–and especially with the Llama model–performance exhibits greater susceptibility to L variations.

3.9 Time Consumption Analysis

In AdaSwitch, the time cost primarily depends on the sequence length and the per-token generation speed. However, in practice, it is difficult to estimate the sequence length for each sample across different KD methods, since generation is performed by different models. Moreover, sequence generation accounts for only a small fraction of KD time. Therefore, we conducted a general and practical experiment by comparing the total runtime of

	SUMM		GSM	
	Llama	Qwen	Llama	Qwen
Off-policy	0.77x	0.79x	0.61x	0.57x
On-policy	1.00x	1.00x	1.00x	1.00x
SKD	1.28x	1.31x	1.61x	1.53x
SWITCH	1.29x	1.30x	1.59x	1.55x
AdaSwitch	1.20x	1.22x	1.41x	1.38x

Table 2: Relative runtime cost of different KD methods over the distillation process. Results are normalized by On-policy (set as 1.00x)

different KD methods over the distillation.

The results in Table 2 indicate that off-policy consistently achieves the lowest runtime cost, since it doesn't conduct sequence generation. On the other hand, all mixing-based methods are slower than on-policy because they require computing teacher logits during generation and switching between the student and teacher.

Comparing the three mixing-based methods, AdaSwitch demonstrates a more stable and lower overhead, averaging around 1.2× on SUMM and 1.4× on GSM. This demonstrates that the single-switch mechanism enhances time efficiency by eliminating repeated teacher–student alternations, thereby reducing synchronization overhead and redundant forward passes in practice.

4 Related Work

4.1 Small Language Model Optimization

The scaling law suggests that increasing the number of parameters makes a language model more powerful. However, relatively small language models (SLMs) are essential in scenarios that are sensitive to latency or computational costs, such as search engines (Li et al., 2025) and recommendation systems (Deldjoo et al., 2024). Therefore, developing a small language model with satisfactory performance for specific tasks becomes crucial.

Various optimization methods can be employed to create an SLM, including Supervised Fine-Tuning (SFT), Direct Policy Optimization (DPO) (Rafailov et al., 2023), and Knowledge Distillation (KD) (Hinton et al., 2015). However, both SFT and DPO require large amounts of high-quality labeled data (Raghavendra et al., 2025; Wang et al., 2024), which can be a significant limitation. In contrast, KD can leverage data of arbitrary size by utilizing a teacher model, making it more scalable. Moreover, while SFT and DPO generally rely on hard labels, KD has the advantage of using intermediate representations or logits from the teacher model as supervisory signals, providing richer information for training the student model (Kim et al., 2022). Given the advantages and widespread use of KD for training SLMs (Muralidharan et al., 2024; Guo et al., 2025; Yang et al., 2025; Abdin et al., 2024), consequently, our work builds on KD to refine its core mechanism for improved performance.

4.2 Knowledge Distillation for LLMs

In the era of LLM, KD has been adapted to autoregressive models and categorized into off-policy and on-policy methods, inspired by reinforcement learning (Xu et al., 2024). Off-policy KD uses either ground-truth sequences or teacher-generated logits for supervision (Kim and Rush, 2016; Sanh et al., 2019). While this ensures high-quality training data, it often causes a mismatch between the student's training distribution and inference-time behavior, leading to inconsistencies (Agarwal et al.). Conversely, on-policy KD trains the student using its own generated sequences, improving consistency and performance on some tasks. However, the student's initially poor outputs limit sequence quality and thus KD effectiveness (Xu et al.). Prior attempts to combine off- and on-policy KD typically mix data at a coarse level (Lin et al., 2020; Agarwal et al.). Among the two most related approaches, SKD employs speculative decoding guided by teacher supervision to enhance sequence quality (Xu et al.). SWITCH (Koo et al., 2025) alternates between the student and the teacher during sequence generation based on a threshold with a fixed decay rate. However, both methods rely on frequent switching between the student and teacher models during generation, which can lead to overfitting to the teacher and limit performance improvements. In contrast, AdaSwitch mitigates this issue by introducing a more stable and efficient token-level mixing strategy.

5 Conclusion

In this paper, we introduce AdaSwitch, a token-level mixed distillation framework designed to reduce the gap between training and inference while preserving high-quality student outputs. AdaSwitch adopts a two-stage generation process that dynamically switches from student exploration to teacher guidance. Extensive experiments on three tasks, three LLM families, and three distance metrics verify its effectiveness, delivering significant performance gains with acceptable time cost.

6 Limitations

Our primary limitation lies in the constrained scope of our experimental evaluation. We validate AdaSwitch on three tasks and three LLM families, which may not fully capture the breadth of challenges faced in diverse real-world applications. Future work will extend the evaluation to a wider range of tasks and domains to further verify the generality and robustness of the proposed method. Additionally, while AdaSwitch achieves significant performance gains, it introduces an additional computational overhead during the KD process. As shown in our time complexity analysis, AdaSwitch incurs approximately 30% higher time consumption compared to the standard on-policy KD method. However, we consider this trade-off acceptable because the KD process is typically performed offline, and crucially, our method introduces no additional latency or computational cost during the inference phase for the final student model.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

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 *The twelfth international conference on learning representations*.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. Dialogsum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, René Vidal, Maheswaran Sathiamoorthy, Atoosa Kasirzadeh, and Silvia Milano. 2024. A review of modern recommender systems using generative models (gen-recsys). In *Proceedings of the 30th ACM SIGKDD conference on Knowledge Discovery and Data Mining*, pages 6448–6458.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Minsoo Kim, Sihwa Lee, Suk-Jin Hong, Du-Seong Chang, and Jungwook Choi. 2022. Understanding and improving knowledge distillation for quantization aware training of large transformer encoders. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6713–6725, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Jahyun Koo, Yerin Hwang, Yongil Kim, Taegwan Kang, Hyunkyung Bae, and Kyomin Jung. 2025. SWITCH: Studying with teacher for knowledge distillation of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3733–3746, Albuquerque, New Mexico. Association for Computational Linguistics.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2961–2984.

Yuchen Li, Hengyi Cai, Rui Kong, Xinran Chen, Jiamin Chen, Jun Yang, Haojie Zhang, Jiayi Li, Jiayi Wu, Yiqun Chen, and 1 others. 2025. Towards ai search paradigm. *arXiv preprint arXiv:2506.17188*.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. 2020. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6121–6133.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. Compact language models via pruning and knowledge distillation. *Advances in Neural Information Processing Systems*, 37:41076–41102.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Mohit Raghavendra, Junmo Kang, and Alan Ritter. 2025. Balancing the budget: Understanding trade-offs between supervised and preference-based finetuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25702–25720, Vienna, Austria. Association for Computational Linguistics.

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

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Qwen Team. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Maolin Wang, Jun Chu, Sicong Xie, Xiaoling Zang, Yao Zhao, Wenliang Zhong, and Xiangyu Zhao. 2025. Put teacher in student's shoes: Cross-distillation for ultra-compact model compression framework. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 4975–4985.

Tianduo Wang, Shichen Li, and Wei Lu. 2024. Self-training with direct preference optimization improves chain-of-thought reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11917–11928, Bangkok, Thailand. Association for Computational Linguistics.

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

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

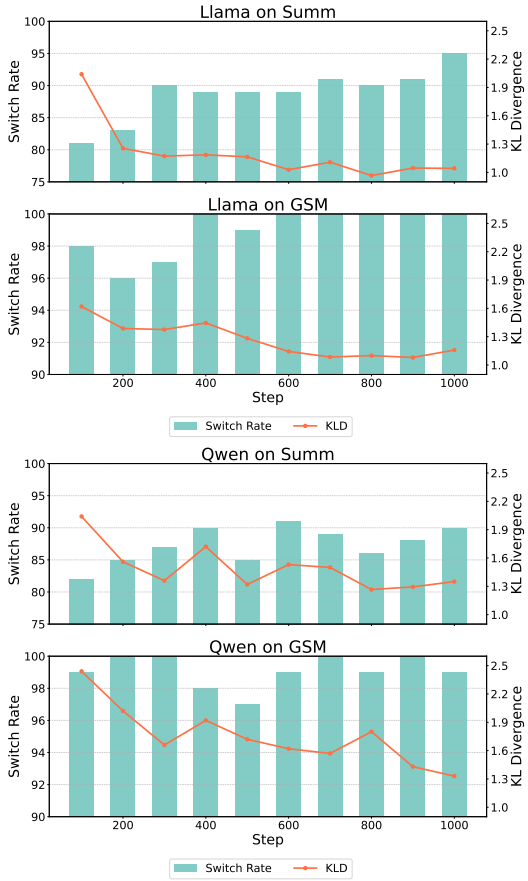

Figure 8: Analysis of the switch rate and KLD between the student and teacher models at the switch token throughout the distillation process. Statistics were collected every 100 steps for the first 1000 steps.

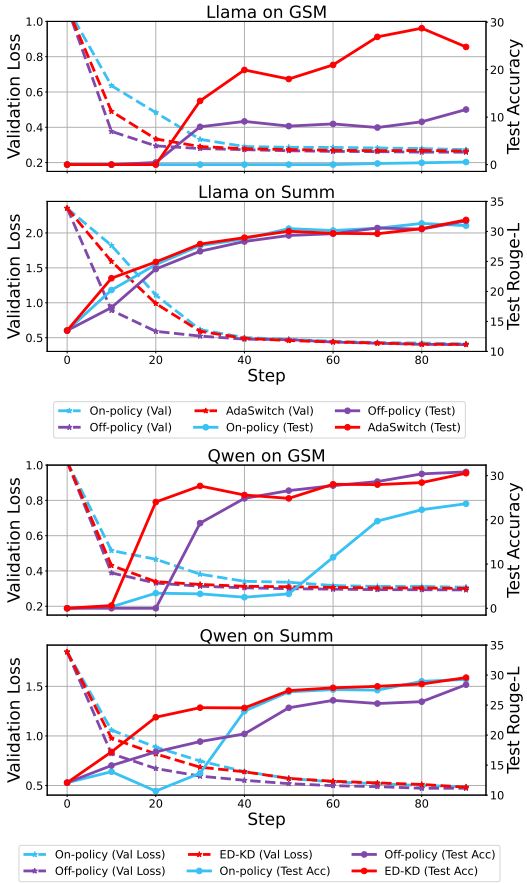

Figure 9: Analysis of the validation loss in early stage of distillation

A Datasets

Dialogue Summarization. (Chen et al., 2021) For the dialogue summarization task, we employ the DialogSum dataset, which contains 12.5K training samples, 500 validation samples, and 1.5K test samples. The complete training set is utilized for SFT of the teacher models. For KD, we randomly sample 1,000 instances from the training data, and the effectiveness of the KD methods is subsequently evaluated on the held-out test set. To assess summary quality, we use the ROUGE-L metric.

GSM8K. (Cobbe et al., 2021) To evaluate arithmetic reasoning, we utilized the GSM8K dataset, a common benchmark for assessing how well language models solve math problems. The dataset comprises 8,500 training and 1,300 test examples, with each example presenting a grade-school word problem alongside its step-by-step solution and final answer. We partitioned the GSM8K training set into a training subset (7,000 instances) for SFT of the teacher models and a validation subset (473 instances) following (Xu et al.). For KD, we randomly selected 1,000 problems from the training data and then evaluated the effectiveness of our KD methods on the separate test set. Accuracy is applied to assess the performance.

GSM-Plus. (Li et al., 2024) GSM-Plus is an adversarial dataset that evaluates the robustness of large language models on mathematical reasoning. This challenging benchmark modifies the original GSM8K problems with numerical, arithmetic, and textual perturbations to test if models can maintain correct reasoning despite subtle changes. In our experiments, we evaluated the student models, which were distilled on GSM8K, on the GSM-Plus test set to further assess the generalizability and robustness of our knowledge distillation methods. The model's performance is assessed using accuracy.

B More Experimental Results

We provide the complete experimental results for the Analysis of Switch Rate and KLD, as well as the Early-Stage Distillation Analysis, in Figure 8 and Figure 9, respectively.