SKIP: a Self-knowledge-guided Step-wise Preference Learning Framework for Concise Reasoning

Qinhong Lin[a], Yuhao Zhang[a], Yinglun Feng[a], Zhongliang Yang[b,a,*], Linna Zhou[b,a,*]
[a]School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China
[b]QuanCheng Laboratory, Jinan, China
{greenred99, yangzl, zhoulinna}@bupt.edu.cn

***Abstract*—While Chain-of-Thought (CoT) reasoning has been proven to be effective, it often leads to overthinking, resulting in computational overhead, inference latency, and even degraded performance in large language models (LLMs). Existing concise reasoning frameworks significantly compromise accuracy while compressing the length of output. In this paper, we propose SKIP, a self-knowledge-guided step-wise preference learning framework. Starting with lightweight fine-tuning to adjust the model's output style, SKIP introduces a carefully designed knowledge probing mechanism to guide model to output an answer at each reasoning step. Based on the correctness of intermediate steps, we construct preference data that guide the model toward more efficient and correct reasoning by leveraging DPO. Experimental results demonstrate that our method effectively improves reasoning compression while mitigating performance degradation after fine-tuning. Besides, SKIP shows strong generalization ability on out-of-distribution datasets. We further conducted ablation studies on the component parameters of our framework.**

***Index Terms*—consise reasoning, LLM, DPO, CoT.**

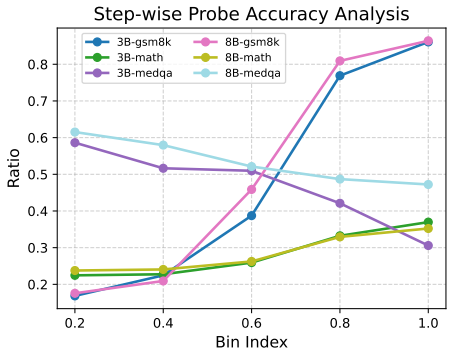

Fig. 1. Step-wise probe accuracy analysis across reasoning progress. Bin Index represents various stages of the inference process, relative to the complete inference. In the GSM8K and Math datasets, the reasoning exhibits positive returns, while in MedQA, it shows negative returns.

I. Introduction

Chain-of-Thought (CoT) reasoning has substantially enhanced the reasoning capabilities of large language models (LLMs), improving their performance on complex tasks such as mathematical reasoning [1]. This advancement holds great promise for enabling LLMs and agents to participate in human decision-making as powerful assistants [2]. Despite these advantages, deeper investigations into the mechanism of CoT reasoning have revealed that LLMs tend to engage in overthinking [3]. This tendency not only incurs additional computational overhead and inference latency but also undermines performance when the model is confident to answer question directly. Reference [4]'s comparative experiments on multiple tasks show that the gains brought by COT are related to the problem type. Empirical studies further show that LLMs exhibit a preference for lengthy answers [5], suggesting that they lack inherent constraints to avoid redundancy or encourage concise reasoning. Thus, various studies have been proposed that focus on compressing reasoning length without compromising model performance [6]. Existing research can be broadly categorized into the following two directions. One is inference-time intervention which tries to add control mechanisms like budget constraints or internal state probes at the model inference process. Another one is model capability elicitation. This line of work aims to encourage concise reasoning through data construction and training objectives. The training approaches, through sophisticated data construction and training, endows the model with the ability to simplify inference, making it more flexible without causing additional inference latency, and this capability can be transferred between different datasets. However, previous work mainly focuses on training models using data guided by context engineering or sampled from more powerful models. This drastic style shift may lead to length compression while severely compromising performance. In this study, we also focus on the latter direction to compress reasoning trace, aiming to tackle three core questions: **Q1:** How to efficiently construct data to elicit the model's ability to concisely compress reasoning? **Q2:** How can we prevent the drop of reasoning accuracy during compression? **Q3:** Can this ability generalize to out-of-distribution (OOD) datasets?

To tackle these questions, we propose leveraging the model's own knowledge to construct efficent datasets for learning which facilitates reasoning compression. Our motivation is driven by a key observation: when models are forced to decode an answer, such as when they encounter the signal 'The answer is', they may be able to generate a correct response in certain situations." As illustrated in Fig. 2, when 'The

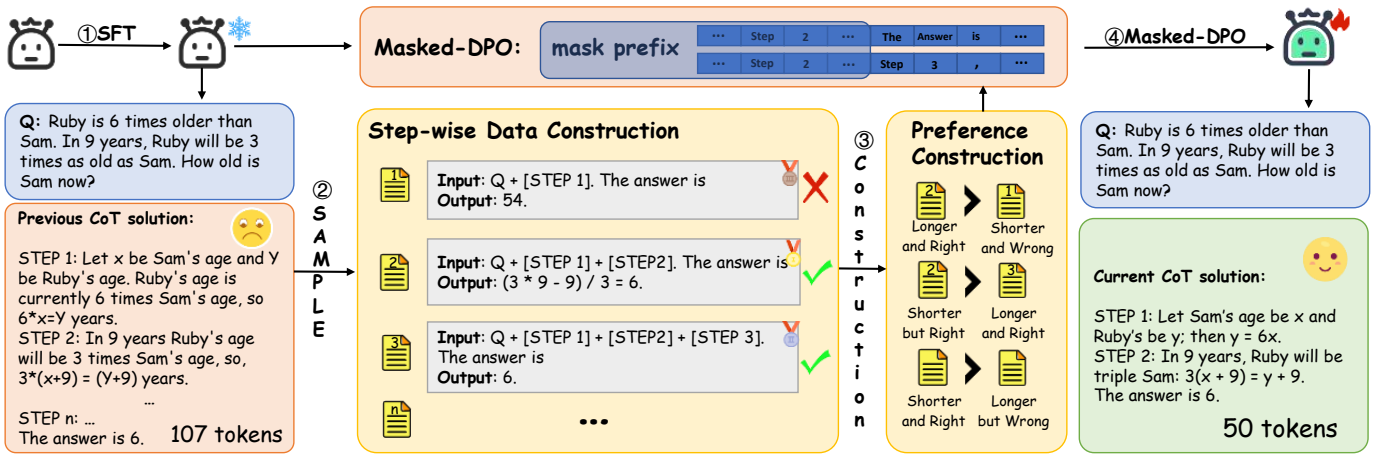

Fig. 2. **S**elf-**K**nowledge-gu**I**ded step-wise **P**reference learning (SKIP), a three stage reasoning compression framework. We use a simple yet efficient probe to detect whether the model can answer the question. We then train models with preference data to teach them to learn when to exit at specific points.

answer is' is inserted at the first step of reasoning and the model is forced to generate an answer, it fails to provide the correct one because the necessary reasoning elements are not yet available. When "The answer is" is inserted at the second step, differently, the model is able to arrive at the correct answer after performing only simple calculations. Without such forced intervention, the model tends to continue producing redundant reasoning steps, thereby increasing the token budget. This phenomenon that some current reasoning paths indeed contain redundant steps, indicates that the model lacks the ability to recognize whether sufficient knowledge has been acquired during reasoning, and thus fails to terminate the reasoning process at the appropriate stage. As shown in Fig.1, we analyzed the step-wise probe accuracy on GSM8K, MATH, and MedQA with Llama3.1-8B-instruct and Llama3.2-3B-instruct. On reasoning benchmarks (GSM8K, MATH), the likelihood of a correct response correlates positively with CoT length, though with diminishing returns in the final stages. In stark contrast, the experimental results on MedQA exhibit a negative trend, with accuracy deteriorating as reasoning proceeds, as models lack sufficient domain knowledge. These findings underscore the necessity of a self-knowledge probing mechanism capable of dynamically identifying the sufficiency of reasoning to determine the optimal early-exit boundary. In this work, we propose a **S**elf-**K**nowledge-gu**I**ded step-wise **P**reference learning (SKIP) framework. At a high level, it's a SFT-DPO two-stage training framework. We first design a simple yet efficient probe to identify model inference state after cold start and construct reasoning compression data for LLMs without relying on external knowledge. We then carefully mix sampled contrastive data to guide the model through preference learning, enabling the compression of reasoning paths without degrading reasoning performance. Experiments on both In-Distribution (ID) and Out-Of-Distribution (OOD) datasets show that the proposed method not only shortens reasoning length effectively but also improves the reasoning capability of LLMs. Our main contributions are as follows:

- We propose a model-free reasoning boundary probing mechanism that efficiently determine whether the model already has the ability to output the correct answer.
- We propose a data construction scheme that balances knowledge preference and length preference, which can be used to train models to improve inference length compression capabilities while mitigating performance degradation under direct preference optimization training.
- Extensive experiments across different tasks and models prove the robustness of SKIP. Besides, we conduct ablation studies on the training data to disentangle and analyzed its specific impact on model performance.

Our data/code is available at https://github.com/linqinhong/SKIP.

II. Related Work

Chain-of-Thought Reasoning. Chain-of-Thought (CoT) [7] reasoning has proven to be a transformative paradigm for achieving human-like cognition and solving complex tasks, as exemplified by the success of reasoning-centric models such as OpenAI o1 [8], Gemini-3.0-Pro [9], and DeepSeek-R1 [10]. During model training, CoT is integrated with techniques like Best-of-N (BoN) [11] and Monte Carlo Tree Search (MCTS) [12], along with the variant, Tree-of-Thought (ToT), to systematically explore the distribution space and sample high-quality synthetic data. Furthermore, the paradigm shift in Reinforcement Learning (RL) from outcome-based rewards [11] to process-based rewards [13] highlights how step-by-step reasoning facilitates deeper cognition, pushing model applications into more specialized and sophisticated domains.
Concise Reasoning. Concise Reasoning aims to mitigate the prohibitive computational overhead and the "overthinking" problem. One primary approach involves inference-time interventions to manage computational budgets. For instance, soft control methods, such as TALE [14], inject budget constraints into the prompt to guide the model toward shorter reasoning

paths. In contrast, hard control strategies, such as [15], [16], employ a prober to predict whether explicit reasoning is necessary for a given task based on the internal states. Another line of work focuses on inducing more concise reasoning styles within the model itself. For instance, FS-BoN [17] leverages few-shot examples to demonstrate brevity, while C3OT [18] utilizes proprietary APIs to distill and compress reasoning steps. These methods typically involve fine-tuning the model to shift its inherent reasoning trajectory toward conciseness. Other studies such as [19]–[21] design length-constrained reward functions within Reinforcement Learning (RL) frameworks to penalize excessive verbosity.

Building on these insights, our work leverages internal state probing to facilitate the construction of high-quality training data. We introduce an explicit mechanism to identify knowledge-sufficient states, enabling the model to strategically bypass redundant reasoning trajectories when its internal information is already adequate.

III. Methodology

As illustrated in Fig. 2, the SKIP framework consists of three stages. First, we leverage a small set of answers with concise reasoning to supervised fine-tune (SFT) the model to learn a more succinct response style. Next, we employ a step-wise knowledge probing mechanism to determine whether the model has already acquired sufficient knowledge to produce the correct answer, and leverage this assessment as guidance for preference-based data construction. Finally, we apply masked-DPO training to inject this capability into the model.

A. SFT Cold Start

Unaligned models often produce answers with inconsistent formats. To address this issue, we first perform SFT cold-start training with two objectives: (i) encouraging the model to output a final summary answer and terminate unnecessary reasoning whenever "The answer is" appears at the end of a reasoning step, and (ii) guiding the model to adopt a more concise response style, thereby providing a better foundation for subsequent DPO training. Specifically, we begin with $N = 8$ question–answer pairs that end with a "The answer is" format as few-shot conditions, and employ in-context learning (ICL) [22] to elicit the model to generate additional pairs in the same style. We filtered data according to correctness and then fine-tune the model via SFT to internalize this reasoning style. In practice, we found that this approach effectively achieves the intended objectives, but also reveals a trade-off between efficiency and reasoning capability: while the reasoning path is compressed, the model's reasoning performance may experience slight degradation.

B. Self-Knowledge-Guided Preference Data

Inference Boundary Probing. After supervised fine-tuning, the model becomes more inclined to output the final answer following a "The answer is" signal. The key to effectively reducing the model's reasoning length while minimizing performance degradation lies in the model's ability to recognize when the generated reasoning path is complete. To achieve this, we employ a simple and effective probe to guide the model within its sampling space. Specifically, during the reasoning process, we use a probe to guide the model to decode and output an answer based on its existing reasoning. So, we sample candidates $Y = \{y_0, y_1, ...y_i\}$ for the question Q. Each candidate can be formulate as:

$$y_i = M(Q, t_0, t_1, ..., t_{i-1}, \boxed{probe}) \tag{1}$$

where i denotes the timestep and t denotes the token. Greedy decoding serves as a reliable proxy for the model's explicit reasoning ability. We approximate a greedy-decoded answer as a representation of the model's internal state. We then assess the correctness of y_i to determine whether the model has acquired sufficient reasoning elements (without confusion, all answers guided by probes are generated through greedy decoding). Therefore, producing the correct answer in this setting indicates that the model has acquired a stable answering capability. Technically, inserting probes at any position is permissible. However, considering that a sentence is the smallest effective unit of the model's reasoning chain, we only insert probes after each sentence. Therefore, the input format to the model can be represented as: "$Q\backslash n[Step_1]\backslash n[Step_2]\backslash n...\backslash n[Step_i]\backslash n\boxed{probe}$". The $\boxed{probe}$ can be any phrase that prompts the model to output an answer. We use 'The answer is' to align with the cold start template, ensuring output quality.

Though models almost always produce a correct answer when the preceding reasoning steps are already logically sufficient, it's important to note that the ability to decode the correct answer at an early step does not guarantee that subsequent reasoning steps will remain correct.This may be because the model, in some cases, has already completed the reasoning implicitly in its internal representations, and may introduce instability or cause logical errors to accumulate when generate additional explicit steps [23]. Please refer to App. C.

Step-wise Preference. Our objective of this stage is twofold: (i) to preserve the reasoning capability of the model as much as possible, and (ii) to encourage the model to terminate reasoning once the existing steps are sufficient to produce the correct answer. As illustrated in Fig. 2, we categorize the sampled answers into four types based on their correctness and length: $longer\ and\ right$, $shorter\ and\ wrong$, $longer\ but\ wrong$ and $short\ and\ right$. In this work, we focus on collecting preference data from two scenarios: knowledge preference data and length preference data. When the model produces a correct answer at $step_i$ but fails to do so at one of $\{step_t | t < i\}$, this indicates that the model has strengthened its reasoning in the additional steps, which represents effective reasoning. We want the model to learn the correct reasoning logic from this. Therefore, we construct knowledge preference data with the relation $longer\ and\ right \succ shorter\ and\ wrong$. When the model successfully produces a correct answer at $step_i$ and also does so at one of $\{step_t | t > i\}$, this indicates that both

reasoning paths are valid. In this case, the extra reasoning in the longer answer should be considered redundant, and we want the model to learn a more concise reasoning pattern from this comparison. Thus, we construct length preference data with the relation $longer\ and\ right < shorter\ but\ right$. In our experiments, we found that the length preference data has a significant impact on the model, leading to a rapid compression of length but also causing a sharp decline in performance. Therefore, we consider a correct answer at $step_i$ only when it is accompanied by an incorrect answer at $step_{t<i}$ and a correct answer at $step_{t>i}$. We analyze this situation in the App. B. Additionally, the reverse case of knowledge preference data, where the longer answer is wrong and the shorter one is correct, suggests that the additional reasoning steps caused an error in an otherwise correct answer. We want the model to terminate the reasoning process in a timely manner. Therefore, we treat $longer\ but\ wrong \succ shorter\ but\ wrong$ as a form of length preference data. It is worth noting that we found the sample size for this case to be very small. It is important to note that we discard the $long\ but\ wrong\ VS\ short\ and\ wrong$ data, as it does not provide effective signals for training.

Masked-DPO. Given x representing the question, y_w representing the preferred data and y_l representing the rejected data,we fine-tuned models using Direct Preference Optimization (DPO) [24] with the following loss:

$$\mathcal{L}(\pi_\theta, \pi_{ref}) = -\mathbb{E}_D[\beta \log \sigma(\log(\frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} / \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}))] \tag{2}$$

where there is $(x, y_w, y_l) \sim D$. During training, we apply a masking strategy to the shared parts of the reasoning steps and question content between the chosen and rejected samples. By training only on the non-overlapping segments, we improve the stability of the optimization process. To improve training efficiency, we fine-tune only the LoRA [25] parameters. After training, we merged LoRA with the models for inference, which introduce no inference latency.

IV. Experiments

Models and Datasets. We conducted experiments with Llama-3.1-8B-Instruct [26], Llama-3.2-3B-Instruct [27] and Qwen-3-14B [28] models on two mathematical reasoning benchmarks that require multi-step reasoning to reach the final solution: GSM8K [29] and MATH [30] and one domain dataset MedQA [31]. To further evaluate the generalization ability of our method, we also performed out-of-distribution (OOD) evaluations on StrategyQA [32] and the Date Understanding task from BIG-Bench Hard [33]. In our experiments, we used the full GSM8K dataset (7,473 training and 1,319 test examples). For MATH, we trained on all 7,500 examples and tested on the first 500. We evaluated StrategyQA on a 500-example subset and BBH's Date Understanding task on 250 test cases.

Baselines. We adopted one inference-time intervention method, Estimated Budget method [14]. For model capbility elicitation methods, we considered the Best-of-N (BON) sampling methods in [17] and the C3OT method in [18]. BON generated training fine-tuning data through few-shot prompting, while C3OT leveraged a stronger closed-source model to compress sampled reasoning trajectories for training. We adopted a prompt-guided CoT approach [34] without any additional control as our default baseline. To better demonstrate the effectiveness of our self-knowledge-guided preference data, we adopted the BON training pipeline as the cold-start setting with $N = 1$ and $N = 8$. While BON utilized the entire training set during SFT, we only used half of the data for SFT and constructed preference pairs from the remaining half. For each setting, we ran the experiments five times and report the mean results.

Evaluation Metrics. Our evaluation metrics included final answer accuracy and average reasoning length. Our goal was to achieve greater reasoning compression with minimal loss in accuracy. In our experiments, we judge answer via regular expression extraction and Exact Match evaluation. We report the consistency of the evaluation results between this approach and LLM-as-a-Judge [35] in the App. A. We also compared the compression efficiency defined as:

$$Eff = \frac{Accuracy}{Average\ Length}, \tag{3}$$

which calculate the accuracy brought by each token.

A. ID results and OOD results

In Tab. IV, We report the results on the in-distribution (ID) datasets. We summarize the key findings as follows:

(1) Our SKIP-8 achieves the highest token efficiency in nearly all experiments, except for one setup where it ranks second. All finetuning methods effectively compress the reasoning length of LLMs compare to the Estimated Budge method. Moreover, in all experiments, both SKIP-1 and SKIP-8 show significant improvements compared to BON-1 and BON-8 with similar training data sizes. Our results show that SKIP framework directly address **Q1:** How to efficiently construct data to elicit the model's ability to concisely compress reasoning? We show a case study in Appendix C.

(2) Two baselines other than BON are also able to reduce reasoning length. However, this comes at the cost of a significant drop in accuracy. By incorporating knowledge preference data, SKIP not only compresses reasoning more effectively but also improves accuracy, providing a positive answer to **Q2:** How can we prevent the drop of reasoning accuracy during compression? It demonstrates that combining knowledge preference data with length preference data is a simple, black-box, yet efficient method.

(3) C3OT could achieves a higher compression ratio than BON methods in some settings, particularly on the GSM8K and Math dataset with Llama-3.1-8B. We refer it to the reason that C3OT leverages API-based sampling with powerful closed-source models to identify redundant reasoning steps during training, thereby achieving strong compression. By contrast, BON relies on in-context learning to guide concise reasoning, which is, to some extent, constrained by the capacity of the base model. This limitation becomes more

TABLE I

RESULTS OF DIFFERENT REASONING COMPRESSION METHODS. THE BOLD TEXT REPRESENTS THE BEST RESULT FOR THE CORRESPONDING METRIC IN THE BLOCK, AND THE TEXT WITH A † REPRESENTS THE SECOND BEST RESULT FOR THE CORRESPONDING METRIC IN THE BLOCK.

model	method	GSM8K			Math			MedQA		
		Acc	Length	Eff	Acc	Length	Eff	Acc	Length	Eff
Llama-3.2-3B	Default	76.70	217.70	0.35	**47.20**	503.20	0.09	56.83	482.36	0.12
	Budget	72.30	141.50	0.51	44.60	438.20	0.10	40.35	339.80	0.12
	C3OT	68.80	115.30†	0.60	39.60	**320.50**	0.12	54.79	101.80	0.54
	BON-1	77.30	149.30	0.52	43.60	392.00	0.11	56.83	99.14	0.57
	SKIP-1	77.70	120.10	0.65†	45.30†	376.30	0.12	58.08	97.47	0.60
	BON-8	78.90†	130.30	0.61	44.20	333.00	0.13†	**58.50**	83.00†	0.70†
	SKIP-8	**79.20**	**114.00**	**0.69**	43.80	321.70†	**0.14**	58.18†	**74.19**	**0.80**
Llama-3.1-8B	Default	**85.50**	240.40	0.36	46.60	479.50	0.10	52.70	278.00	0.19
	Budget	80.40	149.50	0.54	46.20	440.20	0.10	56.51	433.92	0.13
	C3OT	77.20	**109.30**	0.71†	37.40	**302.70**	0.12†	66.30	110.80	0.60
	BON-1	81.60	154.00	0.53	46.30	418.50	0.11	67.50	87.50	0.77
	SKIP-1	82.90	115.20†	**0.72**	**47.80**	398.20	0.12†	67.66†	87.51	0.77
	BON-8	83.40	127.10	0.66	45.10	374.30	0.12†	67.00	76.00†	0.88†
	SKIP-8	84.35†	119.00	0.71†	46.70†	367.90†	**0.13**	**68.90**	**73.10**	**0.94**
Qwen3-14B	Default	88.93	473.57	0.19	70.60	983.48	0.07	59.65	958.50	0.06
	Budget	82.18	359.94	0.23	41.10	855.60	0.05	47.10	685.54	0.07
	C3OT	86.84	**132.84**	0.65†	65.40	700.47	0.09	71.74	439.67	0.17
	BON-1	93.86†	187.48	0.50	76.20	513.25	0.15	72.53 †	255.00	0.28
	SKIP-1	**95.45**	152.47	0.63	**84.40**	359.00†	0.24†	**72.90**	**164.00**	0.31
	BON-8	92.65	161.02	0.58	75.40	487.45	0.15	71.74	224.44	0.32†
	SKIP-8	93.27	138.28†	**0.67**	81.20†	**319.00**	**0.25**	71.80	190.00†	**0.38**

TABLE II

OOD EXPERIMENT RESULTS ON LLAMA-3.1-8B. THE BOLD SCORES DENOTE THE BEST PERFORMANCE.

Train data	Method	BBH(DU)			StrategyQA		
		Acc	Len	Eff	Acc	Len	Eff
	Default	76.8	257.5	0.298	69	303.8	0.227
GSM8K	BON1	66.9	125.2	0.534	51.3	156.5	0.328
	SKIP-1	**72.2**	141.7	0.509	63.1	**124.8**	**0.505**
	BON8	62.4	88.3	0.706	56.3	142.3	0.395
	SKIP-8	67.0	**85.1**	**0.787**	**64.1**	127.9	0.501
MATH	BON1	67.4	147.6	0.456	57.5	241.3	0.23
	SKIP-1	68.0	143.5	0.473	59.8	238.8	0.250
	BON8	69.0	93.6	0.737	65.1	**157.7**	0.412
	SKIP-8	**70.8**	**91.8**	**0.771**	**65.3**	158.0	**0.413**

apparent on challenging datasets like MATH, where C3OT clearly outperforms in terms of compression. However, there is a sharp decline in accuracy with C3OT, indicating that training data from other models may not align well with the model's historical knowledge and capabilities. This suggests that transferring expression styles incurs greater costs. Unlike BON, SKIP utilizes data from distribution sampling, and does not show any significant deterioration in correctness.

In Tab. II, we presented results on out-of-distribution (OOD) datasets. We exclude C3OT and Estimated Budget from this comparison due to their substantial performance degradation on ID datasets, which diminishes the value of extending them to OOD scenarios. The key findings are:

(1) The concise reasoning ability acquired through SFT generalizes effectively to OOD datasets. However, this also comes with a notable drop in accuracy compared to the Default model.

(2) SKIP consistently outperforms BON across all three evaluation metrics. This demonstrates that our constructed preference data not only encourages concise reasoning but also enhances the model's reasoning capability, even in OOD scenarios. These two provide an answer to our research question **Q3:** Can this ability generalize to out-of-distribution (OOD) datasets?

B. Ablation Study

In this section, we analyze the impact of the proposed preference data by examining the effects of data mixing ratios, training data size, and masked-DPO on model performance.

Data mixing ratio In Sec. III-B, we categorize the constructed data into knowledge preference data and length preference data based on forced decoding and the correctness of intermediate reasoning steps. In this section, we explicitly examine their respective effects on model accuracy and reasoning length after training. As shown in Fig. 3(a), increasing the proportion of knowledge preference data in the mixture leads to longer reasoning chains and higher accuracy. Knowledge preference data is derived from detecting incorrect outputs in forced decoding at earlier steps, encouraging the model to learn that sufficient knowledge must be accumulated before producing an answer. In contrast, length preference data plays the opposite role: it guides the model to avoid overthinking once enough knowledge has been acquired. Together, these two types of data form an adversarial learning process that balances reasoning sufficiency with conciseness.

Training data size. In Fig. 3(b), we analyze the impact of training data size on the final performance. Specifically, we use

TABLE III
THE DIFFERENCE BETWEEN DPO AND MASKED-DPO ON LLAMA-3.1-8B. MASKED-DPO COULD ACHIEVE BETTER RESULTS.

Method	DPO-type	GSM8K		MATH	
		Acc	Len	Acc	Len
SKIP-1	DPO	82.18	145	47.1	412
	Masked-DPO	**82.31**	**142**	**47.8**	**401**
SKIP-8	DPO	83.96	119	45.8	367.7
	Masked-DPO	**84.35**	119	**46.68**	**366.9**

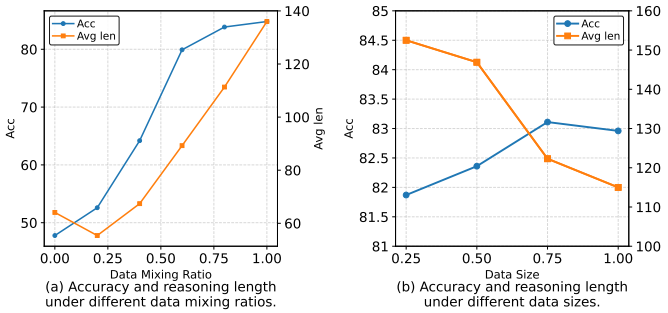

Fig. 3. a.We compared different proportions of knowledge preference data and length preference data to analyze their impact on model performance. b.We analyze the impact of training data size on model performance.

25%, 50%, 75%, and 100% of the half portion for preference data, as described in the experiment setup. As the amount of data increases, the model's accuracy gradually improves while the length decreases steadily with more data. This aligns with the trend in model training and also reflects that our data effectively balances reasoning length and accuracy.

Training Objective. In the constructed step-wise preference data, the chosen and rejected samples share a common prefix in addition to the question itself. To prevent this shared prefix from interfering with training, we apply a masking strategy during DPO training to exclude it. In Tab. III, we compare the two training settings. Experiments on LLaMA-3.1-8B show that masked-DPO yields shorter lengths and higher accuracy.

V. CONCLUSION

In this work, we presented SKIP, a self-knowledge-guided step-wise preference learning framework that leverages intermediate reasoning signals to construct knowledge and length preference data.By combining SFT cold start with step-wise probing, SKIP efficiently detects the model's reasoning critical points and constructs preference data for DPO training, enabling the model to learn more efficient concise reasoning abilities to generate shorter reasoning chains without sacrificing, and even improving, answer accuracy. Our experiments further demonstrate that this capability generalizes effectively to out-of-distribution datasets.

VI. LIMITATIONS

Although we have demonstrated that SKIP exhibits stronger capabilities in concise reasoning, we acknowledge that there are areas in this paper that need further enhancement, which we leave to future work.

More Robust Efficiency Metrics: In this paper, we measure the model's compression efficiency by calculating the accuracy contribution of each individual token. However, we believe that the difficulty of improving model accuracy is influenced by diminishing returns—the closer the model approaches its performance threshold, the harder it becomes to achieve further improvements. The same applies to length compression. Therefore, finding a balance between both factors for a more robust analysis of the model's concise reasoning capability is an important direction for future research.

Broader Evaluation: Although the evaluation in this work covers a broad range, including various types and topics of test questions, it is necessary to apply and evaluate the method on a wider range of data.

ACKNOWLEDGEMENT

This work was supported in part by Research Project of Quan Cheng Laboratory, China (Grant No.QCL20250203)and in part by the National Natural Science Foundation of China under Grant 62172053 and Grant 62302059.

REFERENCES

[1] S. Imani, L. Du, and H. Shrivastava, "Math-prompter: Mathematical reasoning using large language models," in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics(ACL), vol. 5, pp. 37–42, July 2023.

[2] X. Zhang, C. Du, T. Pang, Q. Liu, W. Gao, and M. Lin, "Chain of preference optimization: Improving chain-of-thought reasoning in LLMs," in Advances in Neural Information Processing Systems(NIPS), vol. 37, pp. 333–356, 2024.

[3] X. Chen, J. Xu, T. Liang, Z. He, J. Pang, D. Yu, et al., "Do NOT Think That Much for 2+3=? On the Overthinking of Long Reasoning Models," in Proceedings of the 42nd International Conference on Machine Learning(ICML), 2025.

[4] Z. Sprague, F. Yin, J. D. Rodriguez, D. Jiang, M. Wadhwa, P. Singhal, et al, "To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning," 2024, arXiv:2409.12183. [Online]. Available: https://arxiv.org/abs/2409.12183.

[5] Y. Dubois, B. Galambosi, P. Liang, and T. B. Hashimoto, "Length-controlled AlpacaEval: A simple way to debias automatic evaluators," 2024, arXiv:2404.04475. [Online]. Available: https://arxiv.org/abs/2404.04475.

[6] L. Yue, Y. Du, Y. Wang, W. Gao, F. Yao, L. Wang, et al., "Don't overthink it: A survey of efficient R1-style large reasoning models," 2025, arXiv:2508.02120. [Online]. Available: https://arxiv.org/abs/2508.02120.

[7] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, et al, "Chain-of-thought prompting elicits reasoning in large language models," in Advances in Neural Information Processing Systems(NIPS), vol. 35, pp. 24824–24837, 2022.

[8] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, et al, "Openai o1 system card," 2024, arXiv:2412.16720. [Online]. Available: https://arxiv.org/abs/2412.16720.

[9] M. WILDON, "GEMINI 3.0 PRO 'DEEP THINK'VERSUS AMERICAN MATHEMATICAL MONTHLY PROBLEMS," 2026.

[10] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, et al, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," 2025, arXiv:2501.12948. [Online]. Available: https://arxiv.org/abs/2501.12948.

[11] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, et al, "Learning to summarize with human feedback," in Advances in Neural Information Processing Systems(NIPS), vol. 33, pp. 3008–3021, 2020.

[12] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, "A survey of monte carlo tree search methods," in IEEE Transactions on Computational Intelligence and AI in games, vol. 4, no. 1, pp. 1–43, 2012.

[13] J. Uesato, N. Kushman, R. Kumar, R. Kumar, F. Song, N. Siegel, L. Wang, et al, "Solving math word problems with process-and outcome-based feedback," 2022, arXiv:2211.14275. [Online]. Available: https://arxiv.org/abs/2211.14275.

[14] T. Han, Z. Wang, C. Fang, S. Zhao, S. Ma, and Z. Chen, "Token-budget-aware LLM reasoning," 2024, arXiv:2412.18547. [Online]. Available: https://arxiv.org/abs/2412.18547.

[15] A. Afzal, F. Matthes, G. Chechik, and Y. Ziser, "Knowing before saying: LLM representations encode information about chain-of-thought success before completion," 2025, arXiv:2505.24362. [Online]. Available: https://arxiv.org/abs/2505.24362.

[16] P. Liu, F. Xu, Y. Li, "Token Signature: Predicting Chain-of-Thought Gains with Token Decoding Feature in Large Language Models," 2025, arXiv:2506.06008. [Online]. Available: https://arxiv.org/abs/2506.06008.

[17] T. Munkhbat, N. Ho, S. H. Kim, Y. Yang, Y. Kim, and S. Y. Yun, "Self-training elicits concise reasoning in large language models," 2025, arXiv:2502.20122. [Online]. Available: https://arxiv.org/abs/2502.20122.

[18] Y. Kang, X. Sun, L. Chen, and W. Zou, "C3oT: Generating shorter chain-of-thought without compromising effectiveness," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, pp. 24312–24320, 2025.

[19] Z. Cheng, D. Chen, M. Fu, and T. Zhou, "Optimizing length compression in large reasoning models," 2025, arXiv:2506.14755. [Online]. Available: https://arxiv.org/abs/2506.14755.

[20] Y. Shen, J. Zhang, J. Huang, S. Shi, W. Zhang, J. Yan, N. Wang, K. Wang, Z. Liu, and S. Lian, "DAST: Difficulty-adaptive slow-thinking for large reasoning models," 2025, arXiv:2503.04472. [Online]. Available: https://arxiv.org/abs/2503.04472.

[21] W. Liu, R. Zhou, Y. Deng, Y. Huang, J. Liu, Y. Deng, Y. Zhang, and J. He, "Learn to reason efficiently with adaptive length-based reward shaping," 2025, arXiv:2505.15612. [Online]. Available: https://arxiv.org/abs/2505.15612.

[22] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D Kaplan, P. Dhariwal, et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems(NIPS), vol. 33, pp. 1877–1901, 2020.

[23] Y. Wu, Y. Wang, Z. Ye, T. Du, S. Jegelka, and Y. Wang, "When more is less: Understanding chain-of-thought length in LLMs," 2025, arXiv:2502.07266. [Online]. Available: https://arxiv.org/abs/2502.07266.

[24] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," in Advances in Neural Information Processing Systems(NIPS), vol. 36, pp. 53728–53741, 2023.

[25] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, et al., "LoRA: Low-rank adaptation of large language models," in International Conference on Learning Representations(ICLR), 2022.

[26] Meta Llama Team, "The Llama 3 herd of models," 2024, arXiv:2407.21783. [Online]. Available: https://arxiv.org/abs/2407.21783.

[27] Meta Llama Team, "Llama 3.2 3B Instruct," Sept. 2024. [Online]. Available: https://github.com/meta-llama/llama-models/blob/main/models/llama-3.2/3b-instruct.md. Accessed: Sept. 18, 2025.

[28] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, et al, "Qwen3 technical report," 2025, arXiv:2505.09388. [Online]. Available: https://arxiv.org/abs/2505.09388.

[29] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, et al., "Training verifiers to solve math word problems," 2021, arXiv:2110.14168. [Online]. Available: https://arxiv.org/abs/2110.14168.

[30] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt, "Measuring mathematical problem solving with the math dataset," 2021, arXiv:2103.03874. [Online]. Available: https://arxiv.org/abs/2103.03874.

[31] D. Jin, E. Pan, N. Oufattole, W. H. Weng, H. Fang and P. Szolovits, "What disease does this patient have? a large-scale open domain question answering dataset from medical exams," in Applied Sciences, vol. 11, no. 14, pp. 6421, 2021.

[32] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant, "Did Aristotle use a laptop? A question answering benchmark with implicit reasoning strategies," in Transactions of the Association for Computational Linguistics(TACL), vol. 9, pp. 346–361, 2021.

[33] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, et al., "Challenging BIG-Bench tasks and whether Chain-of-Thought can solve them," in Findings of the Association for Computational Linguistics: ACL 2023, pp. 13003–13051, July 2023.

[34] R. Y. Pang, W. Yuan, H. He, K. Cho, S. Sukhbaatar, and J. Weston, "Iterative reasoning preference optimization," in Advances in Neural Information Processing Systems, vol. 37, pp. 116617–116637, 2024.

[35] L. Zheng, W. L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, et al, "Judging llm-as-a-judge with mt-bench and chatbot arena," in Advances in Neural Information Processing Systems(NIPS), vol. 36, pp. 46595–46623, 2023.

APPENDIX

A. LLM-as-a-Judge VS. Exact Match

In evaluating short answers, LLM-as-a-Judge is increasingly becoming a more effective and accurate approach for assessing answer correctness. Therefore, in Tab. IV, we present the consistency score of the Llama-3.2-3B and Llama-3.1-8B models across three datasets between using both Exact Match (EM) and LLM-as-a-judge as evaluation methods. For each dataset, we randomly sample 300 samples for comparison. The high agreement rate observed between the two methods supports the validity and reliability of our Regex-based EM evaluation pipeline.

TABLE IV
THE MATCH RATIO BETWEEN EM SCORE AND LLM-AS-A-JUDGE ON THE SKIP8 EXPERIMENT SETTING. HIGH CONSISTENCY SUPPORTS THE ACCURACY OF THE EM RESULTS.

Model	Dataset	Consistency
3B	GSM8K	0.91
	Math	0.89
	StrategyQA	0.87
8B	GSM8K	0.94
	Math	0.87
	StrategyQA	0.85

Furthermore, we report the evaluation results when LLM-as-a-judge is employed for data construction and training. The SKIP-1 method on Llama-3.1-8B improves the Best-of-N baseline from 81.6 / 154.0 / 0.53 to 82.1 / 101.7 / 0.807 when trained with data constructed through EM and to 83.47 / 138 / 0.6048 when trained with data constructed through LLM-as-a-Judge. We analyze the lower accuracy scores observed under EM compared to LLM-as-a-judge and attribute this to the fact that EM is prone to generating more False Negatives (i.e., judging a correct answer as wrong due to formatting). Consequently, the LLM-as-a-judge approach successfully identifies more valid preference pairs satisfying "$Longer\ \&\ Right \succ Shorter\ \&\ Wrong$". This enrichment of the training data encourages the model to acquire correct answers through longer, more robust reasoning processes.

B. Length Preference Data

As discussed in Sec. III-B,we define a valid data instance at $step_i$ specifically when it is accompanied by an incorrect answer at $step_{t<i}$ and a correct answer at $step_{t>i}$. Using Llama-3.1-8B on the GSM8K and MATH datasets, we investigate how the inclusion of data satisfying the condition of 'appearing together with an incorrect answer at $step_{t<i}$' affects the final training outcomes.

Our results in Tab. V indicate that while V2-data yields greater length compression and efficiency in most cases, it comes at the cost of a significant decrease in accuracy. We hypothesize that this is because cases that meet the conditions are more likely to contain thoughtful information, while data that does not meet the aforementioned conditions, in some cases due to the questions being too simple, allows the model to arrive at the correct answer without extensive thinking. In this situation, constructing preference data has led the model to learn a certain degree of lazy thinking habits. We argue that in length compression training, simultaneously improving efficiency and accuracy is of greater value. Consequently, we report the results obtained after training with V1-data. Nevertheless, we acknowledge that V2-data is capable of achieving length compression and efficiency that surpass the baseline.

C. Case Study

System: Answer the question.
User: Question: A math teacher had $100 to buy three different types of calculators. A basic calculator costs $8. A scientific calculator costs twice the price as the basic while a graphing calculator costs thrice the price as the scientific. How much change did she receive after buying those three different types of calculators?
Probe at $Step_0$: The answer is 0.
Probe at $Step_1$: First, find the cost of each calculator. The answer is 100.
Probe at $Step_2$: First, find the cost of each calculator.
- Basic calculator: $8 - Scientific calculator: 2 * $8 = $16
- Graphing calculator: 3 * $16 = $48
The answer is 100 - (8 + 16 + 48) = 100 - 72 = $28.
Probe at $Step_3$: First, find the cost of each calculator.
- Basic calculator: $8 - Scientific calculator: 2 * $8 = $16
- Graphing calculator: 3 * $16 = $48
Next, add the costs of the three calculators: $8 + $16 + $48 = $72. The answer is 72.
Probe at $Step_4$: First, find the cost of each calculator.
- Basic calculator: $8 - Scientific calculator: 2 * $8 = $16
- Graphing calculator: 3 * $16 = $48
Next, add the costs of the three calculators: $8 + $16 + $48 = $72. The teacher had $100. Subtract the cost of the calculators from the initial amount: $100 - $72 = $28
The answer is 28.

We present a detailed case study here. By inserting probes at the conclusion of various sentences within a mathematical reasoning chain, we observe that the probes yield correct results at both $step_2$ and $step_4$. Notably, the probe at $step_2$ produces a precise yet more concise logical derivation, which validates the efficacy of our probing setup. In contrast, $step_3$ results in an erroneous output because the preceding sentence disrupts the model's ability to synthesize prior context. When the incorrect response at $step_3$ is paired with the correct one at $step_2$ to construct length preference data, the model learns to truncate reasoning and provide concise answers when the logic is already sufficient. Conversely, when paired with the correct output from $step_4$ as knowledge preference data, the model is encouraged to refine its reasoning until a valid conclusion is reached.

TABLE V
V1 REPRESENTS THE LENGTH PREFERENCE DATA AT $step_i$ CONSTRUCTED UNDER THE CONDITION THAT "THERE IS AN INCORRECT ANSWER AT $step_{t<i}$", WHILE V2 REPRESENTS DATA THAT DOES NOT NEED TO SATISFY THIS CONDITION. THE BOLD TEXT REPRESENTS THE BEST RESULT FOR THE CORRESPONDING METRIC IN THE BLOCK.

Method	GSM8K			Math		
	Acc	Len	Eff	Acc	Len	Eff
BON1	81.6	154.0	0.530	46.3	418.5	0.111
SKIP1-V1	**82.9**	115.2	0.720	**47.8**	398.2	**0.120**
SKIP1-V2	82.1	**101.7**	**0.807**	43.7	**380.6**	0.115
BON8	83.4	127.1	0.656	45.1	374.3	0.120
SKIP8-V1	**84.4**	119.0	0.709	**46.7**	367.9	0.127
SKIP8-V2	82.4	**97.6**	**0.844**	45.4	**338.0**	**0.134**

TABLE VI
COMPARISON OF INFERENCE AND TRAINING DATA VOLUMES BETWEEN BON-8 AND SKIP-8 ON THE LLAMA3.1-8B-INSTRUCT MODEL.

dataset	BON-8		SKIP-8	
	GEN	TRAIN	GEN-SFT/DPO	TRAIN-SFT/DPO
GSM8K	7473*8	7473	3737*8/3736	3737/3169
Math	3750*8	3750	3750*8/3750	3750/2083
MedQA	3000*8	3000	3000*8/3000	3000/1136

D. Training Dataset Statics

We present in Tab. VI a comparison of the inference and training data volumes between BON-8 and SKIP-8 across different datasets, using the Llama3.1-8B-instruct model. Since the preference dataset is constructed by inserting probes during linear generation, each question can be approximately treated as one generation. When constructing preference pairs, a large amount of data is filtered out because it does not satisfy our predefined rules. It can be observed that SKIP-8 has lower overall sampling cost and training cost compared to BON-8.