Self-Distillation for Multi-Token Prediction

Anonymous ACL submission

Abstract

As Large Language Models (LLMs) scale up, inference efficiency becomes a critical bottleneck. Multi-Token Prediction (MTP) could accelerate LLM inference by predicting multiple future tokens in parallel. However, existing MTP approaches still face two challenges: limited acceptance rates of MTP heads, and difficulties in jointly training multiple MTP heads. Therefore, we propose MTP-D, a simple yet effective self-distillation method with minimal additional training cost, which boosts MTP head acceptance rates (+7.5%) while maximumly preserving main-head performance. We also introduce a looped extension strategy for MTP-D, enabling effective and economical MTP head extension and further significant inference speedup to 1-head MTP (+220.4%). Moreover, we systematically explore and validate key insights on the distillation strategies and the potential scalability of MTP through extensive experiments on seven benchmarks. These results demonstrate that our MTP-D and looped extension strategy effectively enhance MTP-head performance and inference efficiency, facilitating the practical usage of MTP in LLMs.

1 Introduction

Large Language Models (LLMs) have demonstrated strong performance across diverse tasks (Guo et al., 2025; Kimi et al., 2025; Qwen, 2025). As task complexity and model scale increase, inference efficiency becomes increasingly important. However, most LLMs rely on the next-token prediction (NTP) paradigm, which performs autoregressive, token-by-token generation and inherently incurs high latency and computational cost, particularly for long sequences (Mehra et al., 2025).

Multi-Token Prediction (MTP) has been proposed as an effective approach to alleviate this inefficiency (Gloeckle et al., 2024). It extends the traditional NTP paradigm by training LLMs with multiple heads, enabling parallel prediction of future tokens. It has been widely adopted in industrial LLMs to accelerate inference (Xiaomi et al., 2025; Qwen, 2025). In general, higher acceptance rates and greater scalability of MTP heads lead to more substantial inference speedups.

The cascaded MTP architecture of DeepSeek-V3 (Deepseek et al., 2024) effectively improves the performance of MTP heads. However, it still faces several challenges for successors: (a) the limited acceptance rates of MTP heads, which could lead to the exponential decline of the cumulative acceptance rate, and thus harm the practical effect of inference speedup. (b) The difficulty in jointly training multiple main and MTP heads. The seesaw effect makes it challenging to jointly improve all heads, while substantial performance decrease of the main head is unaccepted in practice.

To address these issues, we propose **MTP-D**, a simple and effective self-distillation method, which adopts a gradient-detached, Top$N$-logits-selected distillation from the main head to MTP heads in pre-training, along with a looped extension strategy via economical continued pre-training. Specifically, we adopt the Top$N$-selected logits of the main head to guide the training of MTP heads as additional losses, which naturally conforms to the original design intention of MTP with minimal harm to the main head and marginal training cost. Moreover, we propose a looped MTP extension strategy, which takes trained MTP heads as a group, orderly extends them group by group as new MTP heads' initialization, and updates them via continue pre-training. By leveraging intra-group correlations among MTP heads and the distributional consistency induced by distillation, this strategy efficiently extends MTP heads via less tokens, achieving substantial inference speedups.

Experimental results demonstrate that our MTP-D with 4 heads achieves a 7.5% increase in MTP

head acceptance rates with comparable main-head performance, corresponding to a 22.9% speedup. Moreover, our looped extension strategy enables cost-efficient expansion of MTP heads from 4 up to 16, yielding further 35.1% speedup. In addition, we uncover and validate several interesting insights regarding the scalability of MTP. Overall, our key contributions are as follows:

1. We propose MTP-D, a novel self-distillation framework for improving MTP head acceptance rates with comparable main-head performance and less additional cost.

2. We introduce a looped extension strategy, enabling cost-efficient extension of trained MTP heads via continue pre-training.

3. Extensive experiments demonstrate that our method significantly increases MTP head acceptance rates and speedups, shedding light to the practical usage of MTP.

2 Preliminary

Next-Token Prediction is constrained by its training paradigm, which substantially limits both the sample efficiency during pretraining and the inference efficiency of LLMs. Recent studies show that Multi-Token Prediction alleviates these limitations by employing N independent output heads to simultaneously predict the next N tokens, thereby providing richer supervisory signals and significantly improving sample efficiency (Qi et al., 2020). Furthermore, when combined with speculative decoding, MTP can dramatically accelerate LLM inference (Leviathan et al., 2023; Chen et al., 2023; Gloeckle et al., 2024; Li et al., 2024).

DeepSeek-V3 (Deepseek et al., 2024) advances MTP by adopting a sequential, cascaded architecture that explicitly models inter-token dependencies while preserving the complete causal chain and autoregressive nature. This architecture has been widely integrated into prominent industrial LLMs, such as MiMo, GLM, LongCat, and Qwen3-Next (Gloeckle et al., 2024; Xiaomi et al., 2025; GLM4.5 et al., 2025; LongCat et al., 2025; Qwen, 2025). Specifically, the loss function of DeepSeek MTP is formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{mtp}}^{\text{CE}} &= \sum_{k=1}^{K} \alpha_k \, \mathcal{L}_{\text{mtp}_\text{k}}^{\text{CE}} \\ &= \sum_{k=1}^{K} \alpha_k \, \text{CE}\Big(\hat{\mathbf{P}}_{k+1:T+1}^{k}, \, \mathbf{t}_{k+1:T+1}\Big)\end{aligned} \tag{1}$$

where $\text{CE}(\cdot)$ represents the cross-entropy loss, α_k denotes the weight coefficient of the k-th MTP CE loss term, and $\mathbf{t}$ and $\hat{\mathbf{P}}$ denote the ground-truth token sequence and its corresponding predicted probability distribution, respectively.

3 Method

Multi-Token Prediction has been widely proven effective for accelerating LLM inference, which is primarily predicated on the acceptance rate of the speculative tokens generated by MTP heads. We introduce MTP-D, which effectively aligns the top logit distributions of MTP heads with the main head via distillation in pre-training, further enhanced by a looped strategy for economical MTP head extension via continued pre-training.

3.1 Existing Issues of MTP

Currently, practical MTP often faces the following challenges: (a) *Limited acceptance rates of MTP heads.* A persistent performance gap between MTP heads and the main head restricts acceptance rates and thus limits inference acceleration. As shown in Figure 7, MTP heads incur substantially higher losses during pre-training, with losses increasing as the MTP index grows. The cumulative acceptance rate will rapidly drop to unacceptable value even with moderate single head acceptance rates. (b) *Difficulty in jointly training multiple MTP heads.* Adding more MTP heads introduces extra loss terms and hyperparameters, which can hinder main-head optimization and complicate large-scale pre-training. Most practical LLMs have fewer than 1-4 MTP heads in training (Xiaomi et al., 2025).

3.2 Self-Distillation for MTP in Pre-Training

To enhance the MTP heads' performance without compromising the main head's performance, as shown in Figure 1, we propose a gradient-detached, Top$N$-logits-selected self-distillation method. It introduces an additional distillation supervision signal to align the logit distributions of the MTP heads toward those of the main head, which enforce the consistency of logit distributions between the MTP heads and the main head, thereby significantly imporving the performance of the MTP heads.

Specifically, based on the MTP loss $\mathcal{L}_{\text{mtp}}^{\text{CE}}$ in Eq. (1), we introduce a unidirectional Kullback-Leibler (KL) divergence loss (Kullback and Leibler, 1951) for self-distillation from the Top$N$ logits of the main head to the corresponding indices of the

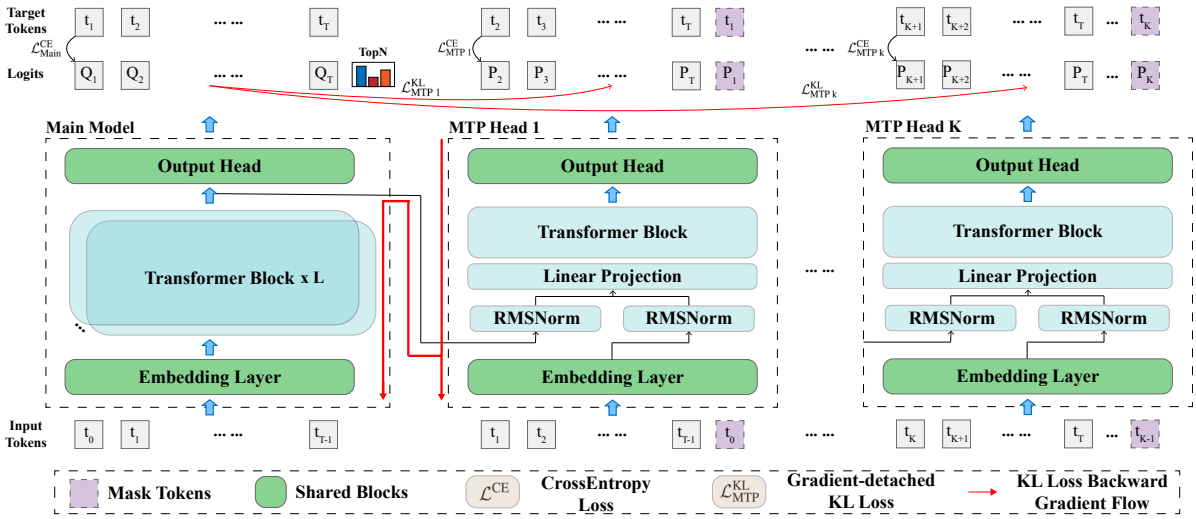


Figure 1: Overview of the gradient-detached, Top$N$-logits-selected self-distillation method.

MTP heads, formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{mtp}}^{\text{KL}} &= \sum_{k=1}^{K} \beta_k \, \mathcal{L}_{\text{mtp}_k}^{\text{KL}} \\ &= \sum_{k=1}^{K} \beta_k \, \text{KL}\left(\tilde{\mathbf{P}}_{k+1:T+1}^{k}, \, \text{sg}(\tilde{\mathbf{Q}})_{k+1:T+1}\right)\end{aligned} \tag{2}$$

where,

$$\mathcal{I}_t^{\text{N}} = \text{TopK}\left(\hat{\mathbf{Q}}_t, N\right) \tag{3}$$

$$\tilde{\mathbf{Q}}_{k+1:T+1} = \sigma\left(\hat{\mathbf{Q}}_{k+1:T+1}\left[\ldots, \mathcal{I}_t^N\right]\right) \tag{4}$$

$$\tilde{\mathbf{P}}_{k+1:T+1}^{k} = \log\left(\sigma\left(\hat{\mathbf{P}}_{k+1:T+1}^{k}\left[\ldots, \mathcal{I}_t^N\right]\right)\right) \tag{5}$$

Here, $\mathcal{I}_t^N$ denotes the set of indices corresponding to the Top$N$ elements of the main head logits $\hat{\mathbf{Q}}_t$ along the vocabulary dimension V for the t-th token. $\sigma(\cdot)$ represents the softmax function, and $\log(\sigma(\cdot))$ corresponds to the log-softmax function. $\text{KL}(\cdot)$ denotes the KL divergence loss, and β_k indicates the weighting coefficient of the k-th MTP KL loss term. Finally, $\text{sg}(\cdot)$ denotes the stop-gradient operation. The core of our method lies in gradient-detached self-distillation and Top$N$-selected logits, as described below.

Gradient-detached self-distillation. In practical settings, we aim to improve the performance of MTP heads while avoiding significant degradation of the main head. To minimize the influence of self-distillation on the main head, we apply a stop-gradient operation to the main head logits $\hat{\mathbf{Q}}$ in $\mathcal{L}_{\text{mtp}_k}^{\text{KL}}$, effectively preventing gradients from propagating back through $\hat{\mathbf{Q}}$. As illustrated by the red path in Figure 1, $\mathcal{L}_{\text{mtp}_k}^{\text{KL}}$ propagates gradients exclusively through $\hat{\mathbf{P}}$ during the backward phase, which is identical to the backward path of the MTP cross-entropy loss $\mathcal{L}_{\text{mtp}_k}^{\text{CE}}$. Consequently, with an appropriate configuration of the weighting coefficients α_k for $\mathcal{L}_{\text{mtp}_k}^{\text{CE}}$ and β_k for $\mathcal{L}_{\text{mtp}_k}^{\text{KL}}$, our method could achieve maximal performance of the MTP heads while maintaining comparable performance of the main head.

Top$N$-selected logits. Modern LLMs often use extremely large vocabularies (e.g., 122,880 in our setting), making full-vocabulary self-distillation loss $\mathcal{L}_{\text{mtp}_k}^{\text{KL}}$ computationally expensive. As Figure 6 shows, main-head logits follow a long-tailed distribution after softmax, with most token probabilities near zero. Directly distilling all logits leads to redundancy, high memory usage, numerical instability, and weak supervision from low-probability tokens. Analyses and ablations indicate that selecting Top$N$ $= 10{,}000$ tokens ensures efficient and stable self-distillation. Details are in Appendix C.

Other explorations. We further investigated several potential self-distillation strategies for MTP, including: (1) different ensemble strategies of the main head and MTP heads as teachers; (2) variants of the KL loss function, including forward, reverse, and hybrid KL; (3) different Top$N$ logits selection strategies, including the Top$N$ of $\hat{\mathbf{Q}}_t$ and the union of the Top$N$ from $\hat{\mathbf{Q}}_t$ and $\hat{\mathbf{P}}_t$; and (4) dynamic adjustment of α_k and β_k with training steps. Subsequent experiments provide a detailed analysis of the effectiveness of each strategy in Section 4.

Final strategy. The final loss for the MTP heads consists of two components: $\mathcal{L}_{\text{mtp}}^{\text{CE}}$ in Eq. 1 and

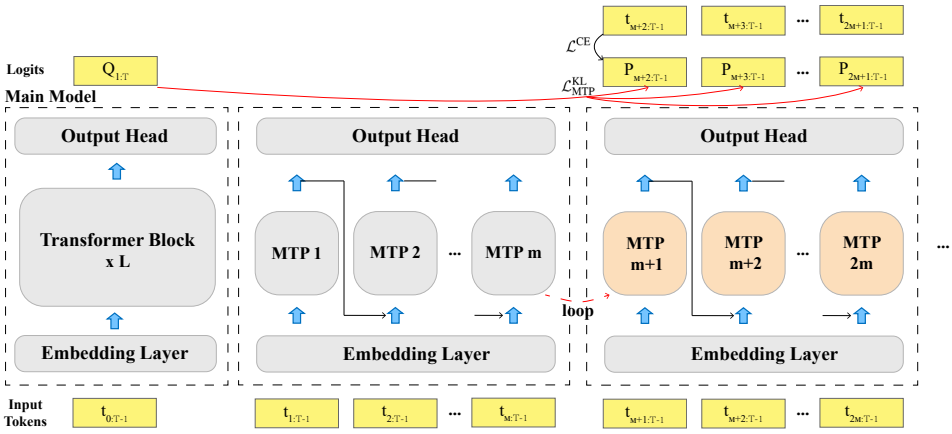


Figure 2: Illustration of the training strategy for looped extension of MTP-D. The gray blocks represent the frozen main model and the trained MTP heads from 1 to m. The weights of the MTP heads from 1 to m are copied to initialize the MTP heads from $m+1$ to $2m$. The orange blocks denote the trainable MTP heads.

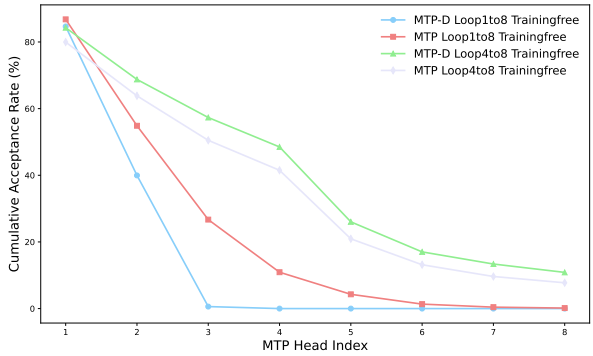
$\mathcal{L}_{\text{mtp}}^{\text{KL}}$ in Eq. 2. $\mathcal{L}_{\text{mtp}}^{\text{CE}}$ aligns the MTP heads with the ground-truth tokens, ensuring the fundamental correctness, particularly during the early stages of pre-training. In contrast, $\mathcal{L}_{\text{mtp}}^{\text{KL}}$ enables the MTP heads to acquire higher-level semantic knowledge from the main head through knowledge distillation, while constraining their probability distributions to be consistent with that of the main head. Together, these two loss terms jointly ensure a stable and effective pre-training: $\mathcal{L}_{\text{mtp}} = \mathcal{L}_{\text{mtp}}^{\text{CE}} + \mathcal{L}_{\text{mtp}}^{\text{KL}}$.

3.3 Looped MTP Head Extension in Continue Pre-Training

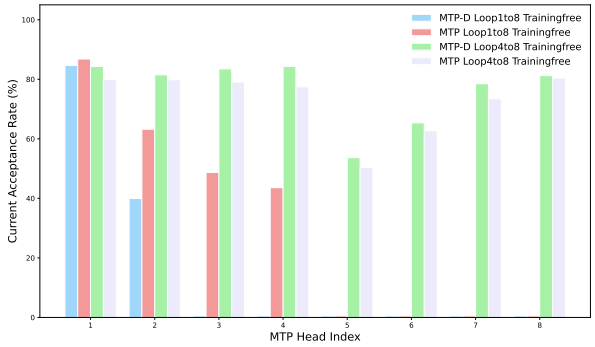
Our MTP-D achieves higher acceptance rates via distillation, thereby establishing a solid foundation for scaling up the number of MTP heads. However, this scaling introduces additional losses and hyper-parameters, which may interfere with the optimization of the main head due to conflicts.

From Figure 1, we observe that the DeepSeek MTP architecture inherently exhibits strong structural consistency and input-output similarity, which provides a natural foundation for scaling up MTP heads via looped extension. As shown in Figure 2, our proposed "**looped extension**" denotes an operation in which a previously trained set of m MTP heads is used to initialize the next set of m heads, followed by continued pre-training on the extended MTP heads. Iteratively repeating this operation allows for a gradual expansion of MTP heads.

Training-Free Explorations. To validate this insight, we conduct a set of pilot experiments under a training-free setting, where both the DeepSeek MTP and our MTP-D are looped up to 8 MTP heads. As illustrated in Figure 3, several key observations can be made as follows:



(a) Cumulative acceptance rates for AGIEval-en.



(b) Acceptance rates for AGIEval-en.

Figure 3: Acceptance rate and cumulative acceptance rate of MTP heads for different models under training-free looped extension. Using the AGIEval-en benchmark as an example, (a) shows the cumulative acceptance rate as the loop is extended to 8 MTP heads for different models, while (b) presents the acceptance rate of each MTP head.

First, MTP heads at loop connection points exhibit a noticeable drop in acceptance rate, though still at an acceptable level. As the number of MTP

heads increases, their acceptance rates gradually recover and approach those of the previously trained heads. These observations suggest that the cascaded architecture of DeepSeek MTP, along with its structural consistency and input-output similarity, inherently supports scalability.

Compared to DeepSeek MTP, MTP-D exhibit substantially better scalability. Under the 1-to-8 loop setting, the cumulative acceptance rate of MTP drops to 0.6% at MTP head 3, whereas MTP-D maintains 26.70%, enabling further scaling. This improvement stems from enhanced consistency of output distributions between MTP heads and the main head, which significantly boosts scalability. Detailed analysis of the training-free looped MTP is in Appendix F.

Looped Extension of MTP-D. The looped extension inherently preserves the correlations among intra-group MTP heads. As shown in Figure 2, continued pre-training is performed using the same self-distillation strategy, which largely maintains consistency between the output distributions of all heads. Additionally, to avoid degrading the performance of the main head and to reduce training cost, both the main model and the previously trained MTP heads are kept frozen.

4 Experiments

4.1 Experimental Setup

Model configuration and training details. Our MTP-D was implemented on both 2B Dense and N10BA1B MoE LLMs. All experiments were conducted on the FineWeb-Edu-350BT dataset (Penedo et al., 2024), where MTP-D pre-training used 350B tokens and looped extension continued pre-training used 70B tokens. Primary experiments were run on 640 NVIDIA H20 GPUs. More details are provided in Appendix B.

Evaluation benchmarks. We evaluate our MTP-D on a diverse set of widely used pre-training benchmarks spanning multiple domains, including AGIEval-en (Zhong et al., 2024) for human-centric standardized examinations; GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) for mathematical reasoning; NaturalQuestions (Kwiatkowski et al., 2019) for knowledge-intensive question answering; SimpleQA (Wei et al., 2024) for short fact-seeking queries; SuperGPQA (Du et al., 2025) for graduate-level knowledge and reasoning; and TriviaQA (Joshi et al., 2017) for reading comprehension.

Speculative decoding and metrics. We adopt a main-head-constrained speculative decoding strategy (Cai et al., 2024; Xia et al., 2023, 2024), which ensures that the inference results of the main model are exactly identical to those obtained with MTP heads, as detailed in Algorithm 1.

Specifically, we report the following metrics: (1) *Accuracy*, defined as the average accuracy of the main head across benchmarks; (2) *Acceptance Rate (AR)*, defined as the percentage of draft tokens generated by each MTP head that are accepted during verification, relative to the total number of tokens verified for that head; (3) *Cumulative Acceptance Rate (CAR)*, defined as the percentage of draft tokens accepted for each MTP head relative to the total number of tokens generated by that head; and (4) *Speedup Ratio*, defined as the relative inference speedup compared to the baseline model, namely the DeepSeek MTP with a single MTP head. Details are provided in Appendix D.

4.2 Main Experiments

Across multiple pre-training benchmarks, we conduct a comprehensive evaluation of the DeepSeek MTP (as the main competitor) and our MTP-D on 2B Dense and A1B MoE LLMs. The evaluation metrics include the average main-head accuracy, the acceptance rate and cumulative acceptance rate of different MTP heads, and the speedup ratio.

(a) ***MTP-D largely improves the acceptance rate of MTP heads while maintaining comparable main-head performance.*** From Table 1, the results demonstrate that our MTP-D substantially improves the acceptance rates of MTP heads with different dense/MoE backbones and model sizes, verifying the effectiveness of our self-distillation. From Table 5 that reflects the main-head performance, for $K = 1$, MTP-D achieves slightly higher average accuracy (11.68) compared to MTP (11.28), and remains comparable when $K = 4$ (10.96 vs. 11.06).

(b) ***MTP-D achieves significantly faster inference speed.*** Table 1 shows that, for $K = 1$, MTP-D improves the (cumulative) acceptance rate by 3.6% over MTP, corresponding to an approximate 14% inference speedup. For $K = 4$, using the cumulative acceptance rate of the 4-th MTP head as an example, MTP-D yields a 7.5% increase, translating to a speedup of 22.9%. Compared to the single-head configuration, the four-head MTP-D achieves a speedup of even up to 107.4%, highlighting the benefits of scaling up MTP heads while noting that achievable speedup is limited by the cumulative

K	Model	Method (MTP)	H	General	Math		Knowledge			STEM
				AGIEval en	GSM8K	MATH	Natural Questions	Simple QA	TriviaQA	Super GPQA
1	2B Dense	MTP	1	85.75	84.86	85.66	90.91	94.28	82.32	85.10
		MTP-D	1	**88.98**	**88.54**	**89.58**	**94.40**	**94.30**	**86.78**	**87.93**
	A1B MoE	MTP	1	85.41	85.24	85.25	91.34	90.65	80.18	82.51
		MTP-D	1	**90.98**	**89.36**	**87.62**	**93.80**	**93.99**	**86.27**	**87.38**
4	2B Dense	MTP	1	81.88 / 81.88	90.32 / 90.32	81.27 / 81.27	89.23 / 89.23	85.17 / 85.17	82.09 / 82.09	82.62 / 82.62
			2	66.81 / 81.59	82.47 / 91.31	64.63 / 79.52	82.55 / 92.52	68.07 / 79.91	66.84 / 81.42	68.26 / 82.62
			3	54.28 / 81.24	74.40 / 90.21	50.83 / 78.65	77.14 / 93.45	55.74 / 81.89	53.39 / 79.89	56.65 / 82.99
			4	45.47 / 83.77	68.52 / 92.09	40.27 / 79.23	**73.51** / 95.30	45.60 / 81.82	43.75 / 81.96	48.62 / 85.83
		MTP-D	1	**85.86** / 85.86	**91.69** / 91.69	**84.73** / 84.73	**90.16** / 90.16	**85.20** / 85.20	**84.71** / 84.71	**86.20** / 86.20
			2	71.96 / 83.81	84.63 / 92.30	69.46 / 81.98	81.15 / 90.01	70.97 / 83.29	71.16 / 84.00	71.96 / 83.48
			3	61.25 / 85.12	78.33 / 92.55	56.75 / 81.70	74.71 / 92.06	57.57 / 81.12	58.57 / 82.31	62.57 / 86.95
			4	**52.96** / 86.46	**72.76** / 92.89	**46.42** / 81.81	71.22 / 95.34	**46.27** / 80.38	**48.52** / 82.85	**54.98** / 87.94
	A1B MoE	MTP	1	82.38 / 82.38	89.63 / 89.63	80.54 / 80.54	82.81 / 82.81	83.64 / 83.64	78.06 / 78.06	80.81 / 80.81
			2	68.11 / 82.67	81.39 / 90.80	63.11 / 78.35	74.54 / 90.02	67.29 / 80.45	61.40 / 78.66	64.48 / 79.79
			3	57.26 / 84.09	74.31 / 91.30	48.58 / 76.98	68.99 / 92.56	52.80 / 78.47	47.66 / 77.62	53.34 / 82.72
			4	48.82 / 85.27	68.22 / 91.81	37.85 / 77.92	65.31 / 94.67	41.87 / 79.28	37.45 / 78.63	45.22 / 84.78
		MTP-D	1	**85.67** / 85.67	**91.41** / 91.41	**82.22** / 82.22	**86.98** / 86.98	**86.40** / 86.40	**82.46** / 82.46	**85.92** / 85.92
			2	71.93 / 83.97	83.84 / 91.72	65.10 / 79.18	79.38 / 91.26	73.20 / 84.72	68.42 / 82.98	72.79 / 84.72
			3	61.55 / 85.58	76.97 / 91.82	49.75 / 76.42	73.24 / 92.26	60.36 / 82.45	55.04 / 80.45	62.88 / 86.38
			4	**52.47** / 85.25	**71.47** / 92.86	**37.89** / 76.16	**68.05** / 92.92	**49.18** / 81.48	**44.51** / 80.91	**54.28** / 86.44

Table 1: Cumulative acceptance rate and acceptance rate of MTP heads for the 2B Dense and A1B MoE models across multiple benchmarks. Here, H denotes the index of the MTP head. For the $K = 1$ group, the CAR is identical to the AR. For the $K = 4$ group, each cell reports CAR/AR (%) values. The best CARs of different MTP heads are highlighted in **bold**. For 2B Dense (A1B) with 1 head, MTP-D improves the average acceptance rate by 3.09% (4.11%). With 4 heads, the improvement is 3.91% (4.73%) for the fourth head.

acceptance rate. Comparisons of the speedup ratios are shown in Figures 4 and 8.

(c) ***MTP-D functions well with different MTP settings, LLM structures, and model sizes.*** MTP-D effectively enhances both acceptance rates and inference speed, with performance gains becoming more pronounced as K increases, and that our method is effective across both dense and MoE. In fact, as the size of the main model increases, the performance improvements brought by distillation to the MTP heads become larger, accompanied by correspondingly greater inference speedups.

4.3 Ablation Study and Model Analysis

We conduct comprehensive ablation experiments to validate the contribution of each component in our MTP-D, using main-head accuracy, MTP-head acceptance rates, and their associated losses as evaluation metrics.

Taking a single MTP head as an example, we conduct ablation studies on four key strategies of MTP-D: `detach`, Top$N$, β_k, and the KL function. Our default configuration is: `detach`, Top$N$ = 10,000, $\beta_k = 1.0$, and forward KL. Analysis is based on the loss curves in Fig. 9 and benchmark results in Table 2.

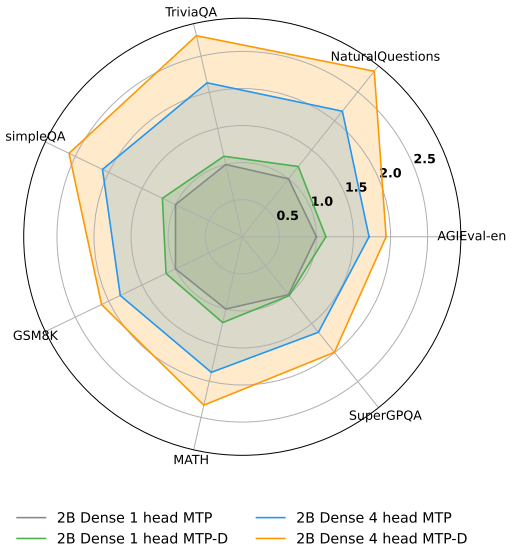


Figure 4: Speedup ratios of the 2B Dense model under different MTP methods and K settings across multiple benchmarks, where the inference speed of 1 head MTP serves as the baseline.

First, removing `detach` greatly increases the main-head loss (+0.079) and reduces its perfor-

Strategy		General	Math		Knowledge			STEM	Mean
		AGIEval en	GSM8K	MATH	Natural Questions	Simple QA	TriviaQA	Super GPQA	
MTP-D		19.20 | 88.98	1.74 | 88.54	1.70 | 89.58	13.00 | 94.30	2.22 | 94.30	37.08 | 86.78	6.81 | 87.93	**11.68** | 90.06
no detach		19.60 | 93.67	1.74 | 93.70	1.50 | 93.78	11.58 | 96.88	2.01 | 96.40	30.42 | 92.31	4.45 | 93.61	10.19 | 94.34
Top$N$	10000x2	18.18 | 89.80	1.44 | 90.12	1.40 | 89.84	13.24 | 94.19	2.41 | 93.74	34.72 | 87.33	7.53 | 88.53	11.27 | 90.51
	1000	20.64 | 90.44	1.36 | 88.63	1.50 | 89.52	12.77 | 94.73	2.29 | 94.06	34.86 | 88.75	6.88 | 87.14	11.47 | 90.47
	1	18.91 | 87.55	1.82 | 87.00	1.65 | 85.76	13.38 | 92.06	2.24 | 92.56	36.11 | 83.11	6.16 | 83.72	11.47 | 87.39
β_k	1.5	20.06 | 91.06	1.67 | 88.95	1.50 | 89.44	14.02 | 94.43	1.99 | 94.52	37.08 | 88.56	5.19 | 88.65	11.64 | 90.80
	0.5	21.17 | 89.03	1.44 | 88.52	1.55 | 88.10	12.71 | 93.31	2.13 | 92.84	35.69 | 87.38	5.59 | 88.86	11.47 | 89.72
	0.3	18.68 | 90.20	2.12 | 85.89	1.75 | 88.19	13.10 | 93.99	1.92 | 91.05	34.72 | 85.50	6.91 | 87.02	11.31 | 88.83
	0.1	18.07 | 88.83	1.74 | 84.69	1.70 | 85.71	12.85 | 92.85	2.18 | 93.18	34.31 | 84.00	6.05 | 84.34	10.99 | 87.66
KL	reverse	22.14 | 89.32	1.67 | 89.06	1.50 | 89.17	12.83 | 94.38	2.06 | 90.99	34.72 | 89.81	6.40 | 88.78	11.62 | 90.22
	hybrid	20.12 | 88.20	1.67 | 87.69	1.55 | 87.84	12.60 | 94.31	1.92 | 92.44	32.36 | 85.67	6.73 | 86.30	10.99 | 88.92

Table 2: Ablation results for MTP-D with 1 head across multiple benchmarks. Each cell shows two numbers separated by "|" (left: main-head accuracy; right: MTP-head AR). The MTP-D configuration is as follows: `detach`, Top$N$ = 10,000, $\beta_k = 1.0$, and the KL function is the forward KL. Underlined entries indicate strategies that significantly reduce main-head performance.

mance (-1.49%), demonstrating that stop-gradient operation on the main-head logits $\hat{\mathbf{Q}}$ effectively prevents gradient backpropagation through $\hat{\mathbf{Q}}$, thereby mitigating interference to the main head.

Next, we examine Top$N$ in the vocabulary dimension V of $\hat{\mathbf{Q}}$. Due to the long-tail distribution in the vocabulary space (Fig. 6), directly distilling over all logits leads to computational redundancy, excessive memory usage, and numerical instability in logarithmic operations, potentially causing gradient oscillation or vanishing (Fig. 10). As Top$N$ increases, the main-head loss slightly rises, but performance remains largely comparable, whereas the MTP-head loss decreases significantly until Top$N$ exceeds a threshold (1,000), beyond which gains plateau. Here, "10000×2" denotes the union of the Top$N$ sets for $\hat{\mathbf{Q}}$ and $\hat{\mathbf{P}}$. Considering the loss factor, we select Top$N$ = 10,000.

The β_k factor is another key strategy. Increasing β_k improves both main-head performance and MTP-head AR, but overly large values significantly increase main-head loss (e.g., $\beta_k = 1.5$, loss +0.028); thus, we choose $\beta_k = 1.0$. Finally, we ablate the KL function (details in Appendix E) and find that forward KL achieves the best trade-off between loss and evaluation performance.

For models with 4 heads, we also perform ablation experiments on the multi-level distillation strategies and the corresponding β_k. Details are provided in Appendix E.2.2.

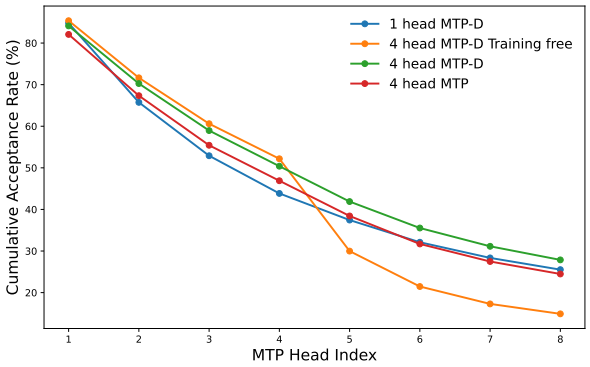
4.4 Results of Looped MTP Extension

In this section, we investigate the loop scalability and upper bounds of models with different numbers of MTP heads, shown in Figures 5. In addition, in Appendix F, Figures 11 and 12 show CAR and AR for MTP head loops up to 8 under a training-free setting. Figures 13 and 14 present CAR for loops up to 8 and 16, respectively, under continued pre-training. Table 8 reports the speedup ratios for various looped extension experiments. From these results, we draw the following insights:

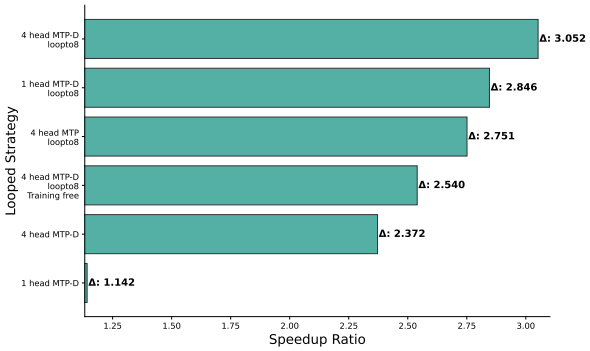
Insight 1: Cascaded MTP is inherently scalable due to the structural consistency and input-output similarity. When extending 4 MTP heads in a training-free manner, the acceptance rate drops mainly at the loop junction (e.g., from 80% to 50%) but remains acceptable, while subsequent heads within the group gradually recover to levels comparable to trained heads.

Insight 2: MTP-D exhibits superior scalability compared to MTP. In training-free extension from 1 to 8 loops, MTP quickly collapses (e.g., CAR drops to 0.6% by the 3-rd head on AGIEval-en), while MTP-D maintains much higher CAR (26.70%). This indicates that distillation improves consistency between MTP heads and the main head, leading to better scalability. Continued pre-training further enhances CAR and speedup across all benchmarks.

Insight 3: Grouped MTP heads exhibit stronger loop scalability than a single MTP head. Compared with single-head extension, grouped



(a) Average cumulative acceptance rate across all benchmarks for different looped extension strategies up to 8 loops.



(b) Average speedup ratio across all benchmarks for different looped extension strategies up to 8 loops.

Figure 5: Comparison of performance across different looped extension strategies with up to 8 loops. The 1-head MTP serves as the baseline.

MTP heads consistently achieve higher CAR and speedup under looped extension, indicating that intra-group correlations are preserved and effectively enhance continued pre-training.

Insight 4: A limited data size is sufficient for looped MTP extension. Scaling the data size from 70B to 350B results in only marginal gains in CAR and speedup ratios for MTP-D.

Insight 5: MTP heads as distillation teachers with the main head enhance head consistency and loop scalability. An ensemble of main- and MTP-head logits achieves comparable CAR while enhancing main-head performance and loop scalability over standard MTP-D.

Insight 6: MTP-D have more potential up to 16 heads. In the 4-to-16 MTP-D setting, the 16th MTP head maintains a CAR of 5–10%. For benchmarks with high acceptance rates (e.g., MATH, SimpleQA, etc), loops up to 16 still provide notable speedup. However, due to the cascaded MTP architecture, CAR gradually declines with more heads, ultimately limiting practical scalability.

5 Related Work

MTP Architectures. The MTP paradigm has been widely studied in both research and industrial LLMs, as it provides rich supervision, improves sample efficiency, and accelerates inference. The classical MTP architecture was introduced by the Meta team (Gloeckle et al., 2024), and DeepSeek-V3 (Deepseek et al., 2024) further proposed the cascaded DeepSeek MTP, which has seen broad adoption. Several studies have explored MTP's mechanisms and applications in LLMs, including adaptation to small models (Aynetdinov and Akbik, 2025), strategies for parallel token prediction (Mehra et al., 2025), and its impact on relational learning (Zhong et al., 2025). Other works investigate MTP variants to enhance training (Grivas et al., 2025; Mahajan et al., 2025) or accelerate inference (Samragh et al., 2025; Liu et al., 2025b; Cai et al., 2025).

Distillation in LLMs. Supervised knowledge distillation (Buciluǎ et al., 2006; Hinton et al., 2015) is a classical technique successfully applied to auto-regressive models (Sanh et al., 2019) and has become central in industrial LLMs, particularly during post-training (Liu et al., 2025a; Agarwal et al., 2024; Gu et al., 2023; Wen et al., 2023). For instance, DeepSeek R1 (Guo et al., 2025) demonstrated that distilling chain-of-thought data enhances reasoning, making distillation a key part of LLM pipelines. It has also been explored in pre-training, with its scaling laws systematically analyzed (Busbridge et al., 2025). Inspired by these advances, we adapt distillation to MTP, where the main head serves as teacher and MTP heads as students, enabling natural self-distillation.

6 Conclusion

In this work, we propose MTP-D, a self-distillation framework for MTP in pre-training, along with a looped extension strategy in continued pre-training. It enables significantly better MTP head acceptance rate and thus faster inference speed with different LLM backbones, while maintaining comparable main-head performance and relatively marginal additional training costs. MTP-D successfully enables up to 8-16 MTP heads with further inference speed gain. Extensive experiments demonstrate the superiority of our methods. The proposed methods and the associated insights provide valuable guidance for improving the pre-training and inference of future LLMs with MTP, as well as practical inspiration for broader applications of MTP.

Limitations

This work focuses on self-distillation for multi-token prediction during pre-training. Future work should explore its adaptation to post-training and investigate the scalability of MTP-D during post-training. Due to resource constraints, the theoretical relationship between optimal α_k and β_k and the number of MTP heads K has not been sufficiently explored, and our method has not been validated on diverse datasets or ultra-large models. We hope it can be validated in industrial-scale LLM pre-training in the future.

References

Kingma DP Ba J Adam and 1 others. 2014. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1412(6).

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*.

Ansar Aynetdinov and Alan Akbik. 2025. Pre-training curriculum for multi-token prediction in language models. *arXiv preprint arXiv:2505.22757*.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russ Webb. 2025. Distillation scaling laws. *arXiv preprint arXiv:2502.08606*.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*.

Yuxuan Cai, Xiaozhuan Liang, Xinghua Wang, Jin Ma, Haijin Liang, Jinwen Luo, Xinyu Zuo, Lisheng Duan, Yuyang Yin, and Xi Chen. 2025. Fastmtp: Accelerating llm inference with enhanced multi-token prediction. *arXiv preprint arXiv:2509.18362*.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

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*.

Aixin Deepseek, Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, and 1 others. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*.

Aohan GLM4.5, Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, and 1 others. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. 2024. Better & faster large language models via multi-token prediction. *arXiv preprint arXiv:2404.19737*.

Andreas Grivas, Lorenzo Loconte, Emile van Krieken, Piotr Nawrot, Yu Zhao, Euan Wielewski, Pasquale Minervini, Edoardo Ponti, and Antonio Vergari. 2025. Fast and expressive multi-token prediction with probabilistic circuits. *arXiv preprint arXiv:2511.11346*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

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*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, and 1 others. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Kimi, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*.

Wanchao Liang, Tianyu Liu, Less Wright, Will Constable, Andrew Gu, Chien-Chin Huang, Iris Zhang, Wei Feng, Howard Huang, Junjie Wang, and 1 others. 2024. Torchtitan: One-stop pytorch native solution for production ready llm pre-training. *arXiv preprint arXiv:2410.06511*.

Kaiyuan Liu, Shaotian Yan, Rui Miao, Bing Wang, Chen Shen, Jun Zhang, and Jieping Ye. 2025a. Where did this sentence come from? tracing provenance in llm reasoning distillation. *arXiv preprint arXiv:2512.20908*.

Xiaohao Liu, Xiaobo Xia, Weixiang Zhao, Manyi Zhang, Xianzhi Yu, Xiu Su, Shuo Yang, See-Kiong Ng, and Tat-Seng Chua. 2025b. L-mtp: Leap multi-token prediction beyond adjacent context for large language models. *arXiv preprint arXiv:2505.17505*.

Meituan LongCat, Bei Li, Bingye Lei, Bo Wang, Bolin Rong, Chao Wang, Chao Zhang, Chen Gao, Chen Zhang, Cheng Sun, and 1 others. 2025. Longcat-flash technical report. *arXiv preprint arXiv:2509.01322*.

Divyat Mahajan, Sachin Goyal, Badr Youbi Idrissi, Mohammad Pezeshki, Ioannis Mitliagkas, David Lopez-Paz, and Kartik Ahuja. 2025. Beyond multi-token prediction: Pretraining llms with future summaries. *arXiv preprint arXiv:2510.14751*.

Somesh Mehra, Javier Alonso Garcia, and Lukas Mauch. 2025. On multi-token prediction for efficient llm inference. *arXiv preprint arXiv:2502.09419*.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, and 1 others. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. *arXiv preprint arXiv:2001.04063*.

Qwen. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Mohammad Samragh, Arnav Kundu, David Harrison, Kumari Nishu, Devang Naik, Minsik Cho, and Mehrdad Farajtabar. 2025. Your llm knows the future: Uncovering its multi-token prediction potential. *arXiv preprint arXiv:2507.11851*.

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*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. F-divergence minimization for sequence-level knowledge distillation. *arXiv preprint arXiv:2307.15190*.

Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. 2023. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3909–3925.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *arXiv preprint arXiv:2401.07851*.

LLM Xiaomi, Bingquan Xia, Bowen Shen, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, and 1 others. 2025. Mimo: Unlocking the reasoning potential of language model–from pretraining to posttraining. *arXiv preprint arXiv:2505.07608*.

Qimin Zhong, Hao Liao, Siwei Wang, Mingyang Zhou, Xiaoqun Wu, Rui Mao, and Wei Chen. 2025. Understanding and enhancing the planning capability of language models via multi-token prediction. *arXiv preprint arXiv:2509.23186*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2299–2314.

A The Use of LLMs

In this paper, we leveraged LLMs to support and refine the writing. Specifically, LLMs were used for grammar and spelling correction, as well as polishing linguistic expressions to enhance clarity and readability.

B Description of Model Configurations and Training Details

Table 3 summarizes the detailed configurations of the two models used in our experiments, namely 2B Dense and 10B A1B MoE. The 2B Dense LLM comprises 32 decoder layers and K MTP heads. The N10BA1B MoE LLM consists of 22 decoder layers (1 "af" layer and 21 "ae" layers) and K MTP heads. Both models employ GQA attention and a vocabulary size of 122,880 tokens. The `mtp_head_layers` is set to 1. Following the DeepSeek MTP architecture, the MTP heads share the embedding layer and output head with the main head. Notably, for both dense and MoE models, a single dense layer rather than a MoE layer is adopted as the MTP head, which is consistent with LongCat (LongCat et al., 2025).

Table 4 reports the training hyperparameters of MTP-D in both the pre-training and continued pre-training stages. These mainly differ in warmup strategy and data size: in continued pre-training, looped MTP disables learning-rate warmup and is trained on 70B tokens. The MTP loss coefficient α_k is set to 0.3 following DeepSeek-V3 (Deepseek et al., 2024), and a fine-grained search is performed for β_k, resulting in $\beta_k = 1.0$ for single-head models ($K = 1$) and $\beta_k = 0.5$ for four-head models ($K = 4$).

Specifically, training employed the AdamW optimizer (Adam et al., 2014) with parameters $(\beta_1, \beta_2) = (0.9, 0.95)$ and $\epsilon = 1 \times 10^{-8}$. A weight decay of 0.1 and gradient clipping with norm 1.0 were applied. The learning rate followed the Warmup-Stable-Decay (WSD) scheduler (Hu et al., 2024), with a maximum of 3×10^{-4} and a minimum of 0. The sequence length was set to 4096 tokens, and the batch size was 8192 tokens. RMSNorm was used for normalization, and rotary position embedding (ROPE) (Su et al., 2024) was applied. For MTP-D, warmup steps were set to 1000 with a total data size of 350B tokens, while for looped MTP, warmup was 0 and the data size was 70B tokens. All experiments were conducted on the FineWeb-Edu-350BT dataset, which is a

randomly sampled subset of approximately 350 billion tokens from the whole FineWeb-Edu corpus (Penedo et al., 2024). Training was conducted using the TorchTitan pretraining framework (Liang et al., 2024). Our primary experiments were conducted on 640 NVIDIA H20 GPUs (98 GB) for approximately 30 days.

Table 3: Detailed description of model configurations.

Hyperparameter	2B Dense	N10BA1B MoE
dim	2048	1536
n_heads	16	32
n_kv_heads	4	4
head_dim	128	128
embedding_tie	False	False
qk_norm	True	True
ffn_hidden_dim	6144	6912
max_seq_len	4096	4096
rope_theta	10000.0	10000.0
norm_eps	1e-5	1e-5
decoder_layers	32 af	af + 21 ae
expert_hidden_dim	-	768
num_experts	-	128
use_shared_expert	-	True
num_shared_experts	-	1
top_k	-	8
use_grouped_mm	-	True
initializer_range	-	0.02
num_mtp_heads	K	K
mtp_head_layers	1	1
mtp_dense_or_moe	dense	dense

Table 4: Training hyperparameter details for MTP-D and looped MTP

Hyperparameter	MTP-D	Looped MTP
Optimizer	AdamW	
Adam (β_1, β_2)	(0.9, 0.95)	
Adam ϵ	1×10^{-8}	
Weight decay	0.1	
Clip grad norm	1.0	
Max lr	3.0×10^{-4}	
Min lr	0	
Lr decay	Cosine	
Decay rate	10%	
Sequence length	4096	
Batch size	8192	
Normalization	RMSNorm	
Vocabulary size	122880	
Positional encoding	ROPE	
Warmup steps	**1000**	**0**
Data Size	**350B**	**70B**
α_k	0.3	
β_k	1.0 for K=1 0.5 for K=4	

C Analysis of Logits Probability Distribution

As illustrated in Figure 6, the main-head logits for the t-th token exhibit a long-tailed distribution after softmax, with most candidate tokens assigned near-zero probabilities. Specifically, the cumulative probability reaches 0.9952 for Top$N$ = 10,000 and 0.8341 for Top$N$ = 1,000. Directly distilling over all vocabulary logits incurs computational redundancy, high memory consumption, and numerical instability, while abundant low-probability signals can hinder effective learning from high-probability ones. Based on these observations, we select Top$N$ = 10,000.

D Speculative Decoding

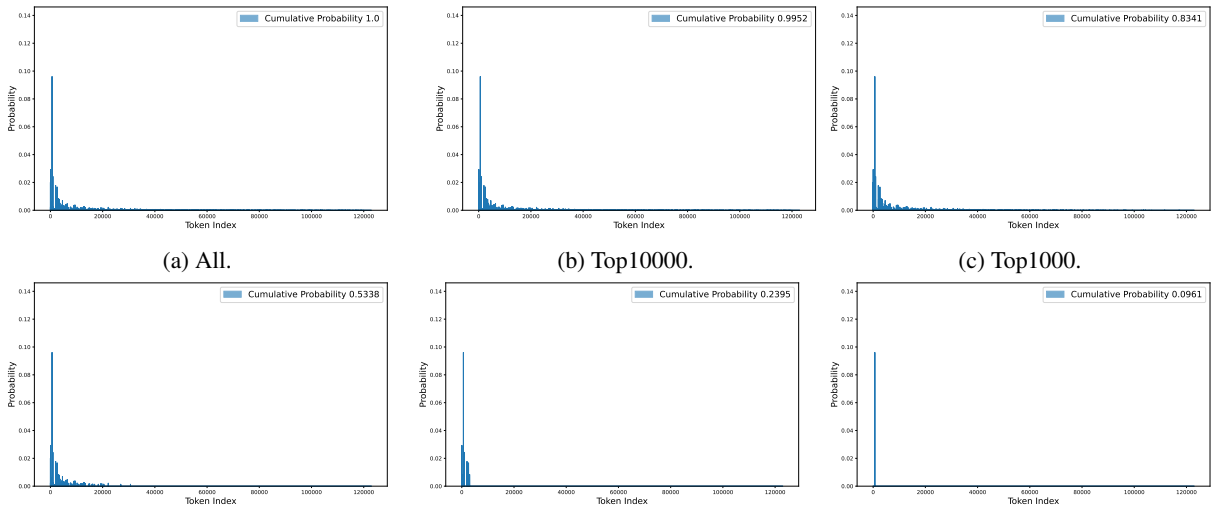
To ensure that the inference results of the main model are exactly identical to those obtained with MTP heads, we adopt a main-head-constrained speculative decoding strategy. To guarantee comparability of inference times across all experiments, inference is performed entirely locally using a single-batch setup, with greedy decoding employed for all samples, and KV cache is not used during inference. Considering the answer lengths in the pretraining benchmarks and the capabilities of the pretraining models, the maximum generation length for each sample is set to 100 tokens.

$$\mathrm{AR}_j = \frac{\sum_{s=1}^{S} A_j^{(s)}}{\sum_{s=1}^{S} C_j^{cmp,(s)}}, \qquad j = 1, \ldots, K. \quad (6)$$

$$\mathrm{CAR}_j = \frac{\sum_{s=1}^{S} A_j^{(s)}}{\sum_{s=1}^{S} C_{\text{step}}^{(s)}}, \qquad j = 1, \ldots, K. \quad (7)$$

Here, S denotes the total number of samples in the corresponding benchmark, and K represents the number of MTP heads. A_j is the number of tokens generated by the j-th MTP head that are accepted by the main head during verification, C_j^{cmp} is the total number of tokens from the j-th MTP head that are verified by the main head, and C_{step} represents the total number of tokens generated by each MTP head, which is the same across all MTP heads.

Notably, due to the pre-training evaluation setup, the maximum generation length is limited to 100. The overall inference speedup from the increased acceptance rate is expected to grow with longer generation lengths, which is particularly important for long-sequence reasoning. In future work, we aim to extend this approach to post-training scenarios.



(a) All. (b) Top10000. (c) Top1000.

(d) Top100. (e) Top10. (f) Top1.

Figure 6: Illustration of the probability distributions of the main head logits for the t-th token under different Top$N$ settings.

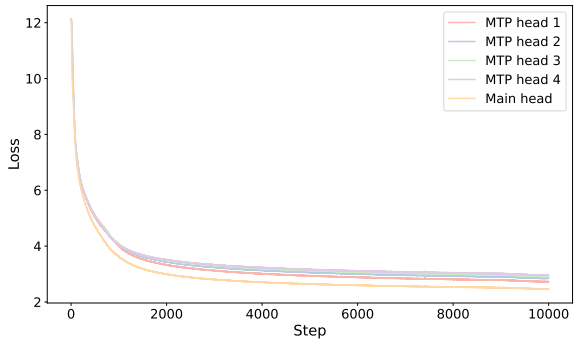


Figure 7: Illustration of the training loss curves for a 2B dense LLM equipped with 4 heads. The final losses of the main head and MTP heads 1 to 4 are 2.47 and [2.73, 2.85, 2.92, 2.96], respectively.

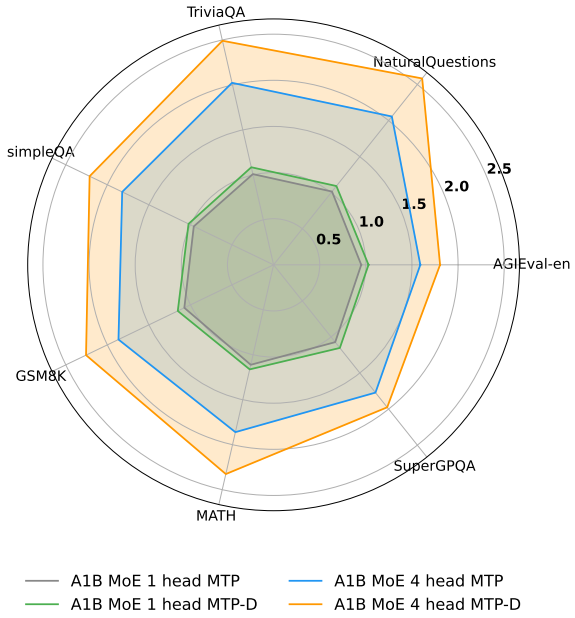


Figure 8: Speedup ratios of the A1B MoE model under different MTP methods and K settings across multiple benchmarks, where the inference speed of 2B Dense DeepSeek MTP with $K = 1$ serves as the baseline.

E Detailed Analysis for MTP-D

The reverse KL loss, defined as $\mathrm{KL}(q\|p) = \mathbf{E}_{\mathbf{x}\sim q}\left[\log \frac{q(\mathbf{x})}{p(\mathbf{x})}\right]$, encourages the model distribution q to align with the modes of the target distribution p. The loss function for the reverse KL is given as follows:

$$\mathcal{L}_{\mathrm{mtp_k}}^{\mathrm{KL,\,R}} = \mathrm{KL}\Big(\mathrm{sg}(\bar{\mathbf{Q}})_{k+1:T+1}, \bar{\mathbf{P}}^{k}_{k+1:T+1}\Big) \quad (8)$$

$$\bar{\mathbf{Q}}^{k}_{k+1:T+1} = \log\left(\sigma\left(\hat{\mathbf{Q}}^{k}_{k+1:T+1}\,[\ldots]\right)\right) \quad (9)$$

$$\bar{\mathbf{P}}_{k+1:T+1} = \sigma\left(\hat{\mathbf{P}}_{k+1:T+1}\,[\ldots]\right) \quad (10)$$

The hybrid KL loss is defined as a weighted combination of the forward KL and reverse KL.

E.1 Details of Main Experiments

Table 5 provides the average main-head accuracy across multiple benchmarks for the main experiments. Figure 8 compares the speedup ratios of different MTP methods using A1B MoE as the backbone.

K	Model	Method (MTP)	General	Math		Knowledge			STEM	Mean
			AGIEval en	GSM8K	MATH	Natural Questions	Simple QA	TriviaQA	Super GPQA	
1	2B Dense	MTP	19.14	0.99	1.55	13.82	2.22	35.14	6.09	11.28
		MTP-D	19.20	1.74	1.70	13.00	2.22	37.08	6.81	11.68
	A1B MoE	MTP	19.56	1.44	1.55	14.56	2.92	48.19	8.05	13.75
		MTP-D	19.35	1.44	1.45	18.14	3.03	48.75	6.88	14.15
4	2B Dense	MTP	19.47	1.29	1.60	13.07	1.97	34.03	5.96	11.06
		MTP-D	19.26	1.74	1.55	12.94	1.83	32.92	6.51	10.96
	A1B MoE	MTP	20.70	1.90	1.55	16.09	2.66	45.83	7.85	13.80
		MTP-D	18.77	1.36	1.45	16.57	2.57	47.36	7.94	13.72

Table 5: Accuracy of the main head for the 2B Dense and A1B MoE models across multiple benchmarks using different MTP methods, showing that the main head of our MTP-D achieves mean accuracy comparable to the baseline.

E.2 Details of Ablation Experiments

In this subsection, we provide a detailed analysis of the ablation experiments for MTP-D with 1 and 4 MTP heads.

E.2.1 1 MTP head

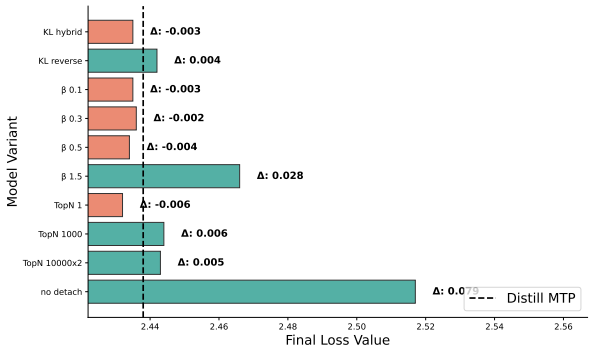
Loss serves as an important metric for pre-training evaluation. Figure 9 presents the final loss comparison of different variants of MTP-D with 1 MTP head, including both main-head and MTP-head losses. While maintaining comparable main-head loss, our MTP-D achieves a substantially lower MTP-head loss than the other variants.

In addition, the loss curves for distillation over the entire vocabulary (Top$N$) are shown in Figure 10. Due to the long-tailed distribution of logits across the vocabulary, logarithmic operations can introduce numerical instability, potentially causing gradient oscillation or vanishing.

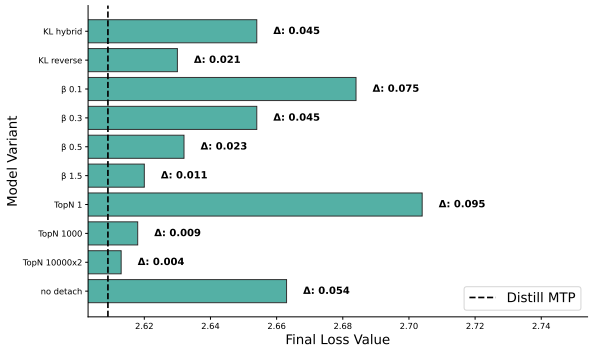
E.2.2 4 MTP heads

We provide additional ablation experiments for MTP-D with 4 heads, analyzing the multi-level distillation strategies and the corresponding β_k values.

For MTP-D with 4 heads, the hyperparameters are set as $\beta_k = 0.5$ (fixed) with main-to-MTP distillation. We first conduct a hyperparameter study on β_k. As shown in Table 6, main-head accuracy initially increases and then decreases with β_k, reaching its peak at 0.3. However, Table 7 shows that the CAR and AR of MTP heads are relatively low for $\beta_k = 0.3$. We further explore a dynamic training strategy where α_k and β_k vary with training steps: for steps < 2000, α_k decreases linearly from 0.7 to 0.3, and β_k increases linearly from 0.1 to 0.5,



(a) Loss comparison of main head.



(b) Loss comparison of MTP head.

Figure 9: Ablation experiments for MTP-D and its variants with loss as metrics. Orange indicates losses better than MTP-D, while green indicates worse performance.

intending for CrossEntropy loss to dominate early training and KL loss to dominate once the main head has sufficient capability. The results, however, do not meet expectations.

Additionally, we investigate more complex distillation strategies based on MTP-D: "ensemble mean" and "ensemble split". In "ensemble mean", the KL teacher for the k-th MTP head is the weighted average of the logits from $\text{MTP}_{0:k-1}$ and the main head.

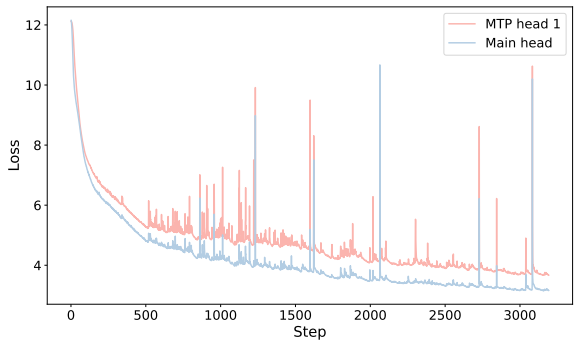


Figure 10: Training loss curves for the 2B Dense model with 1 head using the whole vocabulary.

In "ensemble split", the KL teacher for the k-th MTP head consists of multiple logits from $\text{MTP}_{0:k}$ and the main head, with the losses weighted accordingly. The results indicate that using MTP heads as KL distillation teachers provides more diverse supervision signals, improving both the main model performance and the acceptance rates of MTP heads.

F Detailed Analysis of Looped MTP

In this section, we provide a detailed analysis of acceptance rates and cumulative acceptance rates across multiple pre-training benchmarks for looped MTP, considering both training-free and continued pre-training extensions.

F.1 Training-Free Looped MTP

Figures 11 and 12 show CAR and AR for MTP head loops up to 8 under a training-free setting.

First, MTP heads located at the loop connection points exhibit a noticeable drop in acceptance rate compared to other heads. Taking AGIEval-en for example, the acceptance rate of the 4-to-8 looped MTP drops sharply from approximately 80% at MTP head 4 to around 50% at MTP head 5. Despite this degradation, the acceptance rates still remain at an acceptable level. Moreover, as the MTP heads scale up from 5 to 8, their acceptance rates gradually increase and approach those of the trained MTP heads 1 to 4. A similar trend is observed for the 1-to-8 looped MTP. These results indicate that the cascaded architecture of DeepSeek MTP, together with its structural consistency and input-output similarity, inherently supports scalability.

Furthermore, a comparison between our MTP-D and the DeepSeek MTP under the 1-to-8 loop scaling up setting reveals that MTP heads trained with the self-distillation paradigm exhibit substantially superior scalability. As shown in Figure 3(a), the cumulative acceptance rate of the DeepSeek MTP drops to 0.6% when loop-scaled up to MTP head 3. In contrast, our MTP-D maintains a cumulative acceptance rate of 26.70% at MTP head 3, enabling it to be further scaled up to a larger number of MTP heads. These results demonstrate that our proposed MTP-D significantly enhances the consistency of output distributions across MTP heads with main head, thereby endowing our MTP-D with markedly improved scalability.

F.2 Continued Pre-trained Looped MTP

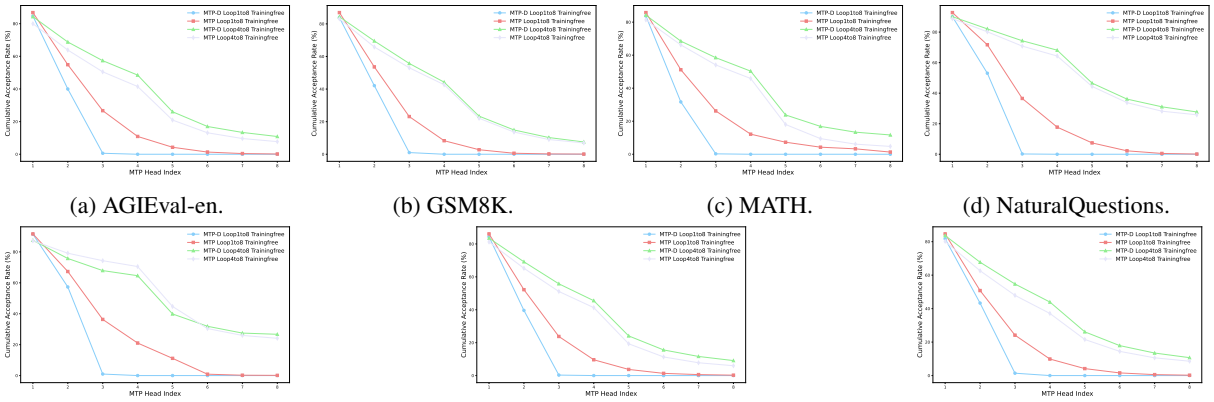
Figures 13 and 14 present CAR for loops up to 8 and 16, respectively, under continued pre-training. Together with Table 8, these results are used to discuss several insights regarding looped MTP in the main text.

Strategy		General	Math		Knowledge			STEM	
		AGIEval en	GSM8K	MATH	Natural Questions	Simple QA	TriviaQA	Super GPQA	Mean
MTP-D		19.26	1.74	1.55	12.94	1.83	32.92	6.51	10.96
β_k	0.1	20.09	1.67	1.6	12.66	2.06	35	3.46	10.93
	0.3	20.88	1.59	1.5	11.75	1.9	33.89	6.38	**11.13**
	1.0	20.06	1.44	1.55	12.33	1.81	31.39	5.74	10.62
step weights		18.63	2.5	1.5	10.41	1.78	26.25	4.64	9.39
ensemble mean		19.03	2.43	1.7	13.13	1.78	33.61	6.9	**11.22**
ensemble split		19.85	1.9	1.65	12.8	1.74	35.42	5.03	11.20

Table 6: Main-head accuracy of ablation experiments for MTP-D with 4 heads across multiple benchmarks. Underlined entries indicate a significant deviation from the main-head accuracy of MTP-D.

Strategy	H	General	Math		Knowledge			STEM
		AGIEval en	GSM8K	MATH	Natural Questions	Simple QA	TriviaQA	Super GPQA
MTP-D	1	85.86 / 85.86	85.20 / 85.20	86.20 / 86.20	91.69 / 91.69	90.16 / 90.16	84.73 / 84.73	87.77 / 87.77
	2	71.96 / 83.81	70.97 / 83.29	71.96 / 83.48	84.63 / 92.30	81.15 / 90.01	69.46 / 81.98	71.16 / 84.00
	3	61.25 / 85.12	57.57 / 81.12	62.57 / 86.95	78.33 / 92.55	74.71 / 92.06	56.75 / 81.70	58.57 / 82.31
	4	52.96 / 86.46	46.27 / 80.38	54.98 / 87.94	72.76 / 92.89	71.22 / 95.34	46.42 / 81.81	48.52 / 82.85
$\beta_k = 0.3$	1	83.57 / 83.57	85.42 / 85.42	83.38 / 83.38	91.39 / 91.39	86.67 / 86.67	82.24 / 82.24	83.86 / 83.86
	2	70.44 / 84.30	70.25 / 82.24	69.96 / 83.91	83.90 / 91.81	78.91 / 91.04	67.02 / 81.49	70.05 / 83.54
	3	60.10 / 85.31	57.33 / 81.61	60.50 / 86.47	75.87 / 90.43	72.06 / 91.32	52.87 / 78.89	57.56 / 82.16
	4	51.85 / 86.28	46.17 / 80.54	52.76 / 87.21	69.91 / 92.14	67.67 / 93.90	42.03 / 79.51	47.37 / 82.30
ensemble mean	1	86.15 / 86.15	86.26 / 86.26	87.05 / 87.05	90.93 / 90.93	86.29 / 86.29	84.99 / 84.99	84.72 / 84.72
	2	73.43 / 85.24	72.44 / 83.97	73.43 / 84.35	83.30 / 91.61	81.12 / 94.00	70.06 / 82.43	71.55 / 84.46
	3	62.84 / 85.58	59.91 / 82.71	65.00 / 88.52	75.86 / 91.07	77.21 / 95.17	57.07 / 81.46	59.47 / 83.11
	4	54.41 / 86.59	49.18 / 82.25	58.25 / 89.63	69.69 / 91.86	73.14 / 94.73	46.57 / 81.60	49.44 / 83.15

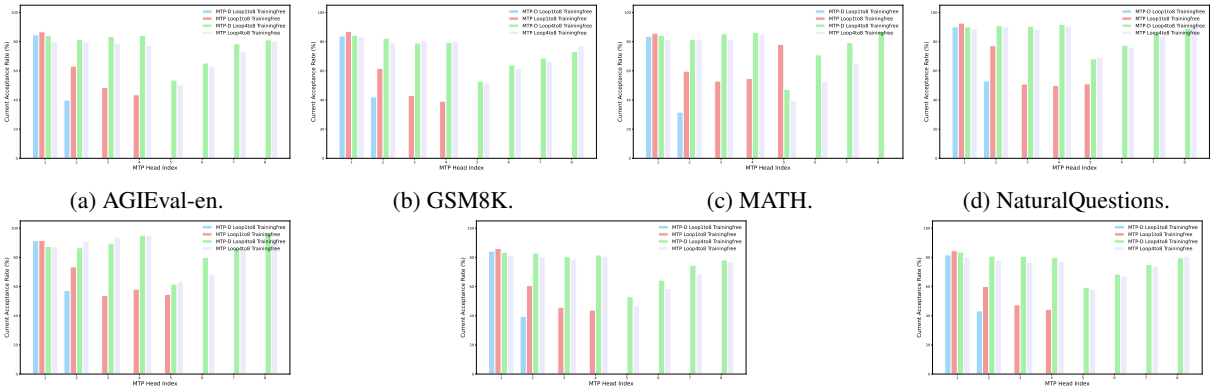
Table 7: MTP-head AR and CAR for MTP-D with 4 heads across benchmarks, comparing three groups with similar main-head performance.



(a) AGIEval-en. (b) GSM8K. (c) MATH. (d) NaturalQuestions.

(e) SimpleQA. (f) SuperGPQA. (g) TriviaQA.

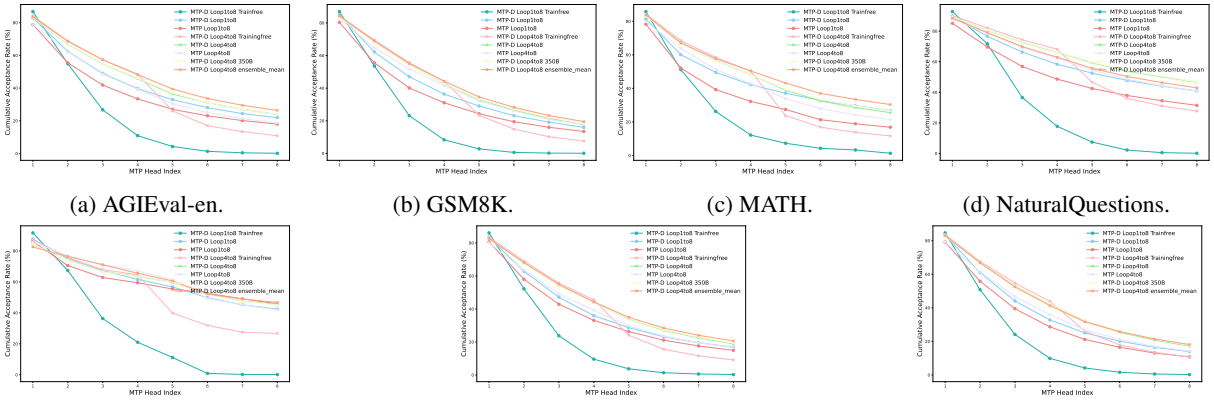
Figure 11: Cumulative acceptance rates on multiple pretraining benchmarks with the training-free looped MTP scaled up to 8.



(a) AGIEval-en. (b) GSM8K. (c) MATH. (d) NaturalQuestions.

(e) SimpleQA. (f) SuperGPQA. (g) TriviaQA.

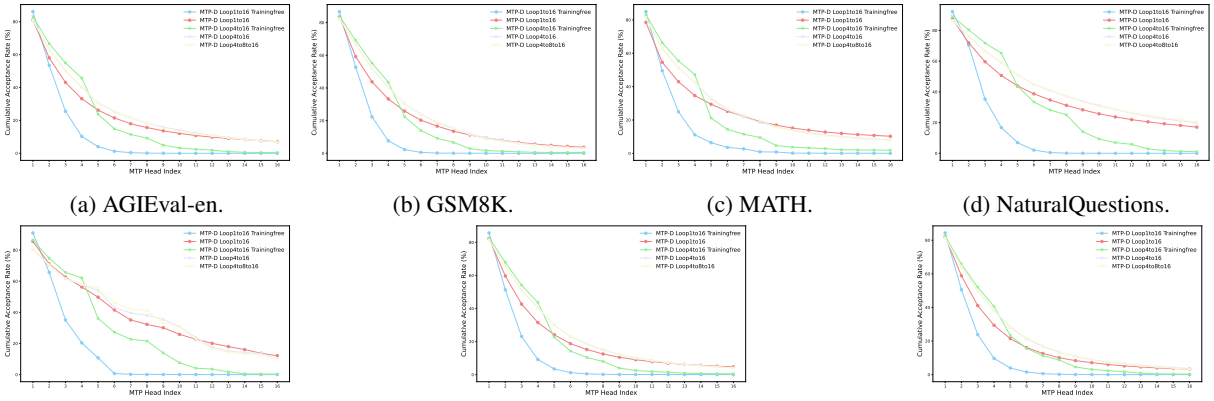
Figure 12: Acceptance rates on multiple pretraining benchmarks with the training-free looped MTP scaled up to 8.



(a) AGIEval-en. (b) GSM8K. (c) MATH. (d) NaturalQuestions.

(e) SimpleQA. (f) SuperGPQA. (g) TriviaQA.

Figure 13: Cumulative acceptance rates on multiple pretraining benchmarks with the looped MTP scaled up to 8.



(a) AGIEval-en. (b) GSM8K. (c) MATH. (d) NaturalQuestions.

(e) SimpleQA. (f) SuperGPQA. (g) TriviaQA.

Figure 14: Cumulative acceptance rates on multiple pretraining benchmarks with the looped MTP scaled up to 16.

		General	Math		Knowledge			STEM	
Loop	**Method**	**AGIEval en**	**GSM8K**	**MATH**	**Natural Questions**	**Simple QA**	**TriviaQA**	**Super GPQA**	**Mean**
1	MTP-D	1.128	1.142	1.188	1.212	1.196	1.114	1.016	1.142
4	MTP-D	1.940	2.107	2.330	2.856	2.594	2.782	1.994	2.372
1 to 8	MTP-D*	1.337	1.451	1.726	2.356	1.971	2.263	1.280	1.769
	MTP	1.764	2.034	2.343	3.835	3.440	3.353	1.733	2.644
	MTP-D	1.964	2.197	2.741	4.193	3.505	3.455	1.865	**2.846**
4 to 8	MTP-D*	1.851	2.058	2.358	3.552	2.885	3.192	1.884	2.540
	MTP	1.870	2.230	2.547	3.962	3.368	3.324	1.951	2.751
	MTP-D	2.071	2.365	2.800	4.573	3.591	3.985	1.980	3.052
	MTP-D 350B	2.068	2.393	2.818	4.593	3.659	3.972	2.096	3.086
	MTP-D ensemble	2.132	2.414	2.972	4.433	3.678	3.959	2.026	**3.088**
1 to 16	MTP-D*	1.062	1.156	1.443	2.452	1.753	2.687	1.050	1.657
	MTP-D	1.535	1.698	2.155	3.586	2.839	3.157	1.571	**2.363**
4 to 16	MTP-D*	1.510	1.713	2.095	4.142	2.806	3.741	1.496	2.500
	MTP-D	1.766	2.012	2.555	5.803	3.884	4.765	1.648	**3.204**
4 to 8 to 16	MTP-D	1.735	1.958	2.465	5.138	3.577	4.323	1.676	2.982

Table 8: Speedup ratios of different loop strategies for MTP-D and MTP during continued pre-training. Except for MTP-D with 350B data sizes, all other continued pre-training experiments use a data size of 70B tokens. $*$ indicates Training Free.

Algorithm 1 Main-Head-constrained Speculative Decoding

Input: Initial prompt sequence $\mathbf{x}$, Main model f, MTP heads $\{g_k\}_{k=1}^{K}$, Max new tokens N

Output: Generated sequence $\mathbf{x}$

```
L_prompt ← |x|    L_gen ← 0
C_step ← 0
for j = 1 to K do
    A_j ← 0  C_j^cmp ← 0
end
while L_gen < N do
    // Verification and Correction
    (P, h) ← f(x)  is_verified ← True
    if L_gen > 0 then
        C_step ← C_step + 1  m ← 0
        for j = 1 to K do
            i ← K − j + 1  C_j^cmp ← C_j^cmp + 1
            x̂ ← Sample(P_{|x|−i})
            if x̂ ≠ x_{|x|−i+1} then
                x ← x_{1:|x|−i} ‖ (x̂)
                h ← h_{1:|x|−i}
                is_verified ← False
                break
            end
            else
                m ← m + 1
            end
        end
        if m > 0 then
            for t = 1 to m do
                A_t ← A_t + 1
            end
        end
    end
    if EOS ∈ x then
        break
    end
    // Speculative Expansion
    if is_verified then
        x_{|x|+1} ← Sample(P_{|x|})
        x ← x ‖ (x_{|x|+1})
    end
    for k = 1 to K do
        (P̂_MTP, h) ← g_k(x, h)
        x_{|x|+1} ← Sample(P̂_MTP)
        x ← x ‖ (x_{|x|+1})
    end
    L_gen ← |x| − L_prompt
end
return x
```