PARD-2: TARGET-ALIGNED PARALLEL DRAFT MODEL FOR DUAL-MODE SPECULATIVE DECODING

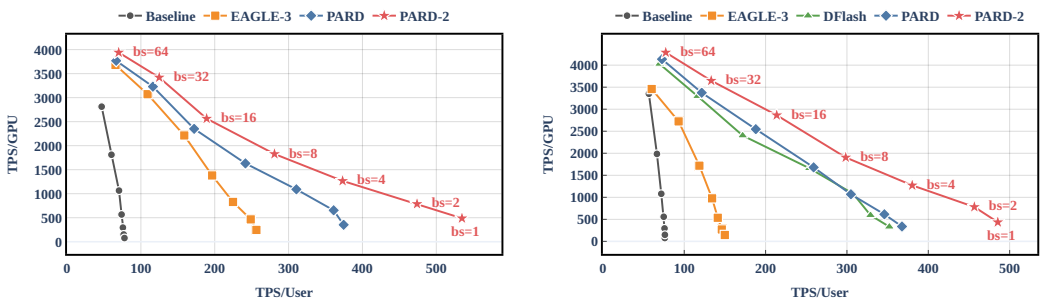
Anonymous authors
Paper under double-blind review

ABSTRACT

Speculative decoding accelerates Large Language Model (LLM) inference by using a draft model to propose candidate tokens that are verified in parallel by the target model. However, existing draft model training objectives are not aligned with the inference-time goal of maximizing consecutive token acceptance. In this paper, we build upon PARD to propose PARD-2, a dual-mode speculative decoding framework. First, we introduce Confidence-Adaptive Token (CAT) optimization, which reweights each token by the cumulative target-model confidence over its prefix, shifting the training focus from per-token accuracy to the acceptance length. Second, through stochastic gating of target hidden features, PARD-2 enables a single draft model to support both target-dependent and target-independent modes, serving an entire model family without retraining. Experiments on Llama3 and Qwen3 show that PARD-2 achieves up to 6.94× lossless acceleration in target-dependent mode and up to 5.71× in target-independent mode.

1 INTRODUCTION

As Large Language Models (LLMs) continue to advance, their capability gains have been accompanied by a rapid growth in model scale, making auto-regressive decoding increasingly expensive at inference time. Speculative Decoding (SD) (Leviathan et al., 2023) has emerged as an effective approach to reducing LLM inference latency: a lightweight draft model proposes multiple candidate tokens, which are then verified in parallel by the target model. A promising line of work trains lightweight drafters conditioned on target-model features, such as Medusa (Cai et al., 2024), Hydra (Ankner et al., 2024), and EAGLE-3 (Li et al., 2025b). However, auto-regressive drafters such as EAGLE-3 require multiple sequential forward passes, incurring non-negligible latency (An et al., 2025; Xiao et al., 2024). To address this, recent works explore parallel drafting: ParallelSpec (Xiao et al., 2024) trains a parallel drafter to generate multiple tokens in a single forward pass, PARD (An et al., 2025) adapts small auto-regressive models for parallel masked-token prediction, and DFlash (Chen et al., 2026) employs a block diffusion model to generate draft tokens in parallel.



(a) Llama3.1-8B (b) Qwen3-8B

Figure 1: **Throughput and Latency Trade-offs on vLLM.** PARD-2 consistently achieves a superior Pareto frontier across various batch sizes (1 to 64) on both (a) Llama3.1-8B and (b) Qwen3-8B.

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

However, existing parallel drafters typically supervise all draft positions uniformly during training, which is suboptimal for both convergence and acceptance length (Liu et al., 2026; Chen et al., 2026). Unlike standard language modeling, which aims to improve prediction accuracy uniformly, speculative decoding is ultimately concerned with how many consecutive draft tokens are accepted by the target model. As illustrated in Figure 2(a), acceptance rates decline consistently at later draft positions, so distant tokens are rarely reached during verification; yet uniform training assigns them equal weight, diverting model capacity from the early positions that matter most. Consequently, naively extending the training draft length can even degrade acceptance length (Table 4), revealing an *inherent limitation of uniformly optimizing all positions*. Recent approaches such as DFlash (Chen et al., 2026) and DART (Liu et al., 2026) mitigate this issue with position-wise decaying weights, but these weights are fixed and depend only on position. Our analysis shows that whether a token is accepted depends not only on its own accuracy but also on whether its entire prefix is accepted; an effective weighting scheme should therefore account for both factors jointly.

In this paper, we introduce PARD-2, a dual-mode speculative decoding framework that mitigates this degradation. We propose **C**onfidence-**A**daptive **T**oken (CAT) optimization, which reweights each token-level objective by a context-dependent confidence score, computed as the cumulative product of the target model's confidence over all preceding tokens in the prefix. This aligns the training objective with the inference-time goal of maximizing the expected acceptance length.

Beyond acceptance length, target dependency remains a barrier for deployment. State-of-the-art drafters such as EAGLE-3 (Li et al., 2025b) and DFlash (Chen et al., 2026) condition on a specific target model's hidden states, so every new target, whether a different size, version, or fine-tuned variant, requires a dedicated drafter. In contrast, target-independent drafters such as PARD (An et al., 2025) can be reused across a model family, but cannot leverage target-side information for stronger acceleration. Existing methods force a trade-off between peak speedup and reusability.

PARD-2 eliminates this trade-off. By stochastically gating target feature injection during training, a single drafter learns to operate both with and without target features: in target-dependent mode, it exploits them for higher acceleration; in target-independent mode, it serves any model in the family without retraining. Unlike EAGLE-3 and DFlash, which rely on grafted target-model layers, PARD-2 retains a standalone architecture and achieves this flexibility without structural overhead.

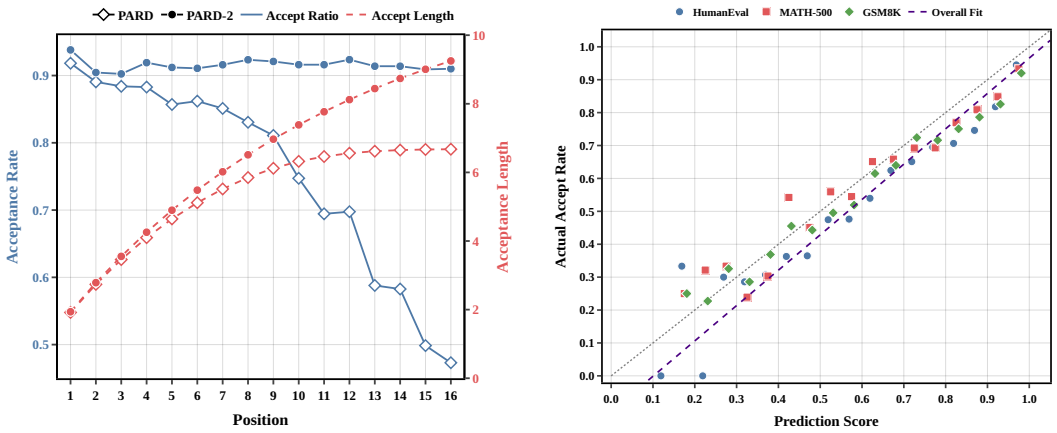
To summarize, our key contributions include:

- We propose PARD-2, a dual-mode speculative decoding framework that, to the best of our knowledge, is the first to unify target-dependent and target-independent drafting within a single draft model. The same drafter can either exploit target hidden states for maximal acceleration or serve as a plug-and-play accelerator for an entire target model family.
- We revisit the objective of speculative decoding and show that a draft token contributes to acceleration only if its entire prefix is accepted. To this end, we propose Confidence-Adaptive Token (CAT) optimization, which reweights each token by the target model's context-dependent confidence over its prefix, improving both prediction and distillation.
- Extensive experiments across diverse models and benchmarks, including deployment in vLLM, show that PARD-2 achieves up to 6.94× lossless speedup and an average 1.3× improvement over PARD. Even in target-independent mode, PARD-2 surpasses the target-dependent DFlash on Qwen3-8B. Furthermore, it delivers the highest throughput under high-concurrency settings, demonstrating strong practical value for real-world deployment.

2 PRELIMINARIES

2.1 SPECULATIVE DECODING

Speculative decoding is a lossless decoding strategy for accelerating LLM inference. Instead of generating each token solely with the target model $\boldsymbol{\theta}_{\text{target}}$, it introduces a smaller and faster draft model $\boldsymbol{\theta}_{\text{draft}}$ to propose multiple candidate tokens in advance, which are then verified by the target model in parallel. This design reduces the number of expensive target model decoding steps while preserving the exact output distribution of the target model.



(a) Position-wise acceptance (b) Confidence vs. acceptance rate

Figure 2: **Acceptance behavior of Llama3.1-8B.** (a) On HumanEval, PARD-2 achieves higher acceptance rates and longer acceptance length than PARD across token positions, mitigating distant-position degradation. (b) Target-model confidence strongly correlates with the actual acceptance rate, supporting its use as a proxy for token-level acceptance.

Formally, given a prefix $X = (x_0, \ldots, x_{n-1})$, speculative sampling uses a lightweight auto-regressive draft model $\boldsymbol{\theta}_{\text{draft}}$ to propose a length of K tokens, denoted by $\tilde{Y} = (\tilde{y}_n, \ldots, \tilde{y}_{n+K-1})$. The proposal probability distribution factorizes as

$$P(\tilde{Y} \mid X; \boldsymbol{\theta}_{\text{draft}}) = \prod_{k=0}^{K-1} P(\tilde{y}_{n+k} \mid x_0, \ldots, x_{n-1}, \tilde{y}_n, \ldots, \tilde{y}_{n+k-1}; \boldsymbol{\theta}_{\text{draft}}) \,. \tag{1}$$

For position $n+k$, let $p_k(y) = P(y \mid x_0, \ldots, x_{n-1}, \tilde{y}_n, \ldots, \tilde{y}_{n+k-1}; \boldsymbol{\theta}_{\text{target}})$ and $q_k(y) = P(y \mid x_0, \ldots, x_{n-1}, \tilde{y}_n, \ldots, \tilde{y}_{n+k-1}; \boldsymbol{\theta}_{\text{draft}})$ denote the target and draft conditional probabilities, respectively. The target model verifies the draft tokens in order, accepting $\tilde{y}_{n+k}$ with probability

$$a_k = \min\left(1, \frac{p_k(\tilde{y}_{n+k})}{q_k(\tilde{y}_{n+k})}\right), \qquad k = 0, \ldots, K-1. \tag{2}$$

The target model accepts the longest valid prefix and, upon the first rejection, samples a correction token from the residual distribution, preserving exact equivalence to sampling from the target model. Ignoring the bonus token, the probability that the first $k+1$ draft tokens are all accepted is $\prod_{j=0}^{k} a_j$. Hence, the expected acceptance length L is

$$\mathbb{E}[L \mid X, \tilde{Y}] = \sum_{k=0}^{K-1} \prod_{j=0}^{k} a_j. \tag{3}$$

2.2 Parallel Draft Models

Although speculative decoding significantly accelerates LLM inference, auto-regressive drafting still requires K sequential forward passes to generate K draft tokens, which limits the end-to-end speedup. To address this issue, recent work has explored parallel draft models that predict multiple tokens simultaneously. DiffuSpec (Li et al., 2025a) and DFlash (Chen et al., 2026) adopt diffusion-based drafters that generate tokens through iterative denoising. To better match the auto-regressive architecture of the target model, PARD (An et al., 2025) retains an auto-regressive backbone and introduces masked placeholders, enabling parallel masked-token prediction in a single forward pass.

In particular, PARD introduces a special mask token m and predicts each future token conditioned only on the prefix and preceding mask placeholders. Its draft probability distribution is

$$P(\tilde{Y} \mid X; \boldsymbol{\theta}_{\text{PARD}}) = \prod_{k=0}^{K-1} P(\tilde{y}_{n+k} \mid x_0, \ldots, x_{n-1}, m_n, \ldots, m_{n+k-1}; \boldsymbol{\theta}_{\text{PARD}}) \,. \tag{4}$$

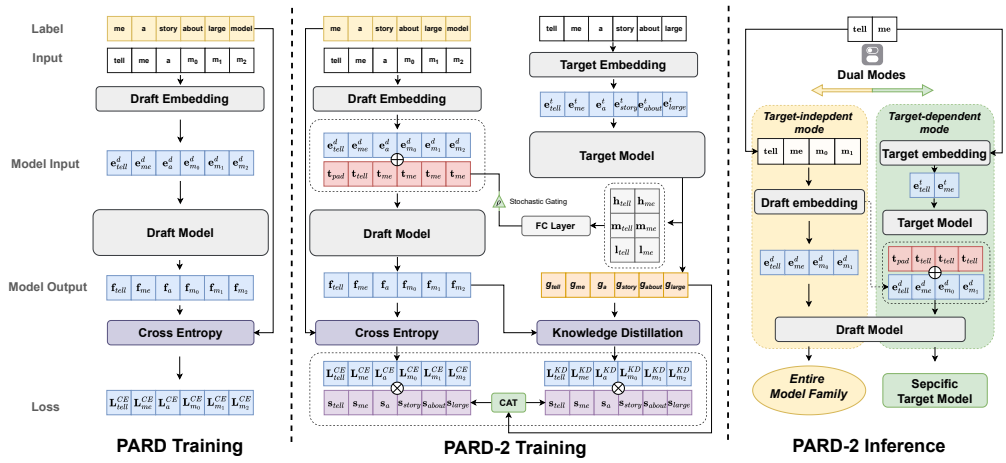


Figure 3: **Overview of PARD-2.** The training (mid) and inference (right) designs of PARD-2. Compared to PARD (left), PARD-2 integrates CAT optimization, target hidden features, and knowledge distillation. PARD-2 supports flexible switching between target dependent and independent modes.

Because each position depends only on the prefix and mask tokens, all K predictions can be computed in a single forward pass. This approach not only substantially reduces drafting latency but also ensures target independence, enabling the drafter to be reusable across a family of target models.

Given the ground-truth $Y = (y_n, \ldots, y_{n+K-1})$, PARD is trained with the cross-entropy loss

$$\mathcal{L}_{\text{PARD}} = -\frac{1}{K} \sum_{k=0}^{K-1} \log P(y_{n+k} \mid x_0, \ldots, x_{n-1}, m_n, \ldots, m_{n+k-1}; \boldsymbol{\theta}_{\text{PARD}}) \,. \tag{5}$$

3 Method

3.1 Observation

The draft length K is a key design choice for parallel draft models. To study its effect, we train PARD with two draft lengths, $K_{train} = 8$ and $K_{train} = 16$. As shown in Table 4, increasing K_{train} yields little improvement and can even degrade performance across several benchmarks. This observation contradicts the common intuition that a longer draft length should naturally translate to a greater acceptance length and enhanced decoding efficiency. To understand this phenomenon, we analyze the verification mechanism of speculative decoding. Because the target model evaluates candidate tokens strictly in order, the acceptance of any subsequent token heavily relies on the successful verification of all its predecessors. Let a_j denote the probability that the j-th draft token is accepted given that all preceding draft tokens are accepted. Eq. equation 3 can be decomposed by position as

$$\mathbb{E}[L \mid X, \tilde{Y}] = \sum_{k=0}^{K-1} \prod_{j=0}^{k} a_j = \sum_{k=0}^{K-1} \left(\prod_{j=0}^{k-1} a_j \right) a_k. \tag{6}$$

This decomposition reveals two key factors that govern whether the token at position k is accepted. The first factor, $\prod_{j=0}^{k-1} a_j$, is the probability that all previous draft tokens are accepted. The second factor a_k is the probability that the current token is accepted once position k is reached.

We then define the first factor as s_k:

$$s_k := \prod_{j=0}^{k-1} a_j, \qquad s_0 := 1. \tag{7}$$

Since the k-th token can contribute to the acceptance length only if all preceding tokens are accepted, s_k can be interpreted as the importance of that token for acceleration. With this notation,

$$\mathbb{E}[L \mid X, \tilde{Y}] = \sum_{k=0}^{K-1} s_k a_k. \tag{8}$$

The second factor, a_k, reflects the local quality of the draft prediction at position k. Since the token-level training objective of the draft model aims to improve prediction quality at each position, it is naturally related to increasing a_k. This insight motivates us to reweight the per-token training objectives by s_k, assigning higher importance to tokens situated on highly probable accepted prefixes.

3.2 Confidence-Adaptive Token Optimization Strategy

Motivated by the above observations, we assign adaptive weights to tokens during training, aligning the optimization objective with the goal of maximizing acceptance length. However, the ideal weight $s_k = \prod_{j<k} a_j$ cannot be computed during training, since each a_j depends on the draft model's evolving predictions and is only available during verification at inference time.

Inspired by EAGLE-2 (Li et al., 2024b) and CAPE (Du et al., 2024), we investigate the relationship between the target model's confidence and the token acceptance rate across multiple benchmarks, including HumanEval, GSM8K, and Math-500. As illustrated in Figure 2(b), the target model's confidence exhibits a strong positive correlation with the empirical acceptance rate. Consequently, we can leverage the target model's confidence scores as a reliable proxy for the expected token acceptance rate, enabling adaptive token-level weighting.

Building upon this empirical finding, we approximate the actual acceptance probability using the target model's confidence on the corresponding ground-truth token y_{n+k} conditioned on its prefix:

$$\hat{c}_k := P(y_{n+k} \mid x_0, \ldots, x_{n-1}, y_n, \ldots, y_{n+k-1}; \boldsymbol{\theta}_{\text{target}}) \,. \tag{9}$$

We then estimate the importance of each token by computing the cumulative product of the target confidences along the prefix:

$$\hat{s}_k := \prod_{j=0}^{k-1} \hat{c}_j, \qquad \hat{s}_0 := 1. \tag{10}$$

Here, $\hat{s}_k$ approximates the probability that position k is reached during verification. Using $\hat{s}_k$ as a stop-gradient weight, we obtain the final PARD-2 training objective:

$$\mathcal{L}_{\text{PARD-2}} = -\frac{1}{K} \sum_{k=0}^{K-1} \hat{s}_k \log P(y_{n+k} \mid x_0, \ldots, x_{n-1}, m_n, \ldots, m_{n+k-1}; \boldsymbol{\theta}_{\text{PARD-2}}) \,. \tag{11}$$

Unlike the uniform loss in Eq. equation 5, Eq. equation 11 adaptively prioritizes tokens that are likely to contribute to the final accepted prefix and reduces the influence of distant positions that are rarely reached during speculative verification. As a result, the resulting objective better matches the inference-time acceleration goal of speculative decoding.

3.3 PARD-2 Training

During training, in addition to assigning different importance weights to each token-level loss to better optimize the speculative decoding objective, we further adapt the draft model to align with the target model, enabling it to generate highly compatible proposals.

Stochastic Gating for Target Features. As illustrated in Figure 3, given an input sequence X, we extract hidden representations from multiple layers of the target model, denoted by l, m, and h, corresponding to low-, middle-, and high-level features, respectively. These hidden states are fused into a compact target-context feature $t = \text{Proj}([l; m; h])$, where $[\cdot;\cdot]$ denotes concatenation and $\text{Proj}(\cdot)$ is a lightweight projection module. To improve training efficiency, we further process the draft-model input with Conditional Drop Token (COD) (An et al., 2025), which selectively drops conditional tokens during training. The fused target hidden feature t is then injected into the draft model by adding it to the draft-model input embeddings e^d. In this way, the draft model can leverage target-side context during drafting, leading to better alignment with the target-model distribution.

Moreover, to achieve target independence, we do not inject target-context features for every training instance. Instead, we stochastically inject them during training: $e^{d'} = e^d + \xi \cdot t, \xi \sim \text{Bernoulli}(1-\rho)$. Thus, $e^{d'} = e^d$ with probability ρ, and $e^{d'} = e^d + t$ otherwise. This design reduces target-side dependence, enabling a single drafter to serve the entire model family.

Training Loss Function. To further improve draft-target alignment, we augment the supervised training objective with knowledge distillation from the target model. Without loss of generality, let $x_{\leq n}$ denote the current prefix and x_{n+k} denote the k-th future token to be predicted. Note that the acceptance probability of a future token depends not only on its relative position k, but also on the current prefix $x_{\leq n}$. Therefore, we denote the estimated token-level acceptance weight as $\hat{s}_{n,k}$, which captures the expected acceptance likelihood of token x_{n+k} conditioned on prefix $x_{\leq n}$. Specifically, for the k-th future position, our objective is formulated as a weighted sum of the standard cross-entropy loss $\mathcal{L}^{\mathrm{CE}}$ and the distillation loss $\mathcal{L}^{\mathrm{KD}}$:

$$\mathcal{L}_k = \sum_{n=1}^{N-k} \hat{s}_{n,k} \left(\beta \mathcal{L}^{\mathrm{CE}}_{n,k} + \mathcal{L}^{\mathrm{KD}}_{n,k} \right), \tag{12}$$

where β balances the supervised and distillation terms. Here, $\mathcal{L}^{\mathrm{CE}}_{n,k}$ and $\mathcal{L}^{\mathrm{KD}}_{n,k}$ are computed for token x_{n+k} given prefix $x_{\leq n}$, and $\hat{s}_{n,k}$ is the confidence-adaptive weight assigned to this token.

3.4 DUAL-MODE INFERENCE

As illustrated in Figure 3, the same PARD-2 drafter supports two inference modes without any architectural change or additional fine-tuning. In target-dependent mode, the multi-layer hidden states of the target model are obtained as a by-product of prefilling and of each verification step. After every verification, the features of the newly accepted tokens are fused into t and added to the corresponding draft input embeddings with the same one-position shift as in training, while the mask tokens reuse the feature of the last verified token. This allows the drafter to exploit up-to-date target-side context for maximal acceleration. In target-independent mode, t is omitted, i.e., $e^{d'} = e^{d}$, which corresponds to the gated-off case seen with probability ρ during training. The drafter thus operates as a standalone model that generalizes across an entire target model family without retraining.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models. We evaluate PARD-2 on the Llama3 (Llama Team, 2024) and Qwen3 (Yang et al., 2025) model families. In target-dependent mode, we train and evaluate on Llama3.1-8B, Qwen3-8B, and Qwen3-14B. In target-independent mode, a single drafter serves Qwen3 models from 8B to 32B.

Datasets and Benchmarks. PARD-2 is trained on the PARD dataset (Magpie (Xu et al., 2024) and Evol-CodeAlpaca (Luo et al., 2023)), augmented with samples from Nemotron-v2 (Basant et al., 2025) and Nemotron-v3 (Blakeman et al., 2025). We evaluate on HumanEval (Chen et al., 2021) for code generation, GSM8K (Cobbe et al., 2021) and MATH-500 (Lightman et al., 2023) for mathematical reasoning, and MT-Bench (Zheng et al., 2023) for multi-turn dialogue.

Metrics. PARD-2 is a lossless acceleration method that preserves the original target model and exact acceptance rule. We focus on acceleration performance and report the following metrics:

- **Speedup**: The acceleration ratio over vanilla auto-regressive decoding.
- **Acceptance Length** τ: the average number of draft tokens accepted in each verification.
- **Tokens Per Second**: The number of tokens generated per second in real-world scenarios.

Implementation Details. For training, we extract target hidden features from 4 layers of the target model. The draft model is trained on AMD MI300X GPUs, utilizing a batch size of 64 and a draft length of $K = 16$. We set $\rho = 0.1$ and the loss weighting coefficient $\beta = 0.1$. For inference, all throughput evaluations are implemented based on the vLLM framework. To ensure a fair comparison, tree-based decoding is explicitly disabled across all methods. Unless otherwise specified, all evaluation experiments are conducted on NVIDIA A100-40GB GPUs. We employ a tensor parallelism degree of TP=2 for Qwen3-32B, while setting TP=1 for all other models.

4.2 EXPERIMENTAL RESULTS

In this section, we evaluate PARD-2 on Qwen3 and Llama3 with thinking mode disabled. We compare PARD-2 with several SD baselines, including EAGLE-3, DFlash, and PARD. For the Qwen3

Table 1: Target-dependent comparison of speedup ratios and average acceptance lengths τ across different methods. Q3 represents Qwen3 model family and L3 represents Llama3 model family.

Target	Method	HumanEval		GSM8K		Math-500		MT-Bench		Average	
Temperature = 0		SpeedUp	τ	SpeedUp	τ	SpeedUp	τ	SpeedUp	τ	SpeedUp	τ
Q3 8B	EAGLE-3	2.11	2.52	1.98	2.36	1.90	2.26	1.71	2.01	1.92	2.29
	DFlash	5.66	7.03	4.64	5.67	5.59	6.88	2.57	3.17	4.61	5.69
	PARD	5.02	6.33	4.91	5.97	4.96	6.28	2.68	3.28	4.39	5.46
	PARD-2	**6.75**	**8.80**	**6.44**	**8.15**	**6.87**	**8.74**	**3.16**	**3.83**	**5.81**	**7.38**
Q3 14B	EAGLE-3	2.32	2.70	2.10	2.44	2.09	2.42	1.72	1.96	2.06	2.38
	PARD	5.16	6.07	5.10	5.88	4.98	5.82	2.78	3.21	4.51	5.25
	PARD-2	**6.74**	**8.60**	**6.40**	**7.93**	**6.73**	**8.47**	**3.37**	**3.90**	**5.81**	**7.22**
L3.1 8B	EAGLE-3	3.34	4.95	2.80	3.77	2.37	3.24	2.54	3.49	2.76	3.86
	DFlash	3.62	4.91	2.78	3.74	2.53	3.46	2.42	3.32	2.84	3.86
	PARD	5.09	6.59	4.20	5.20	3.84	4.90	2.85	3.60	3.99	5.07
	PARD-2	**6.94**	**9.85**	**5.21**	**6.54**	**5.36**	**7.63**	**3.25**	**4.16**	**5.19**	**7.04**

Table 2: Target-independent performance comparison. Each method uses a single checkpoint for all Qwen3 targets, identical to its Qwen3-8B checkpoint in Table 1, without target hidden features.

Target	Method	HumanEval		GSM8K		Math-500		MT-Bench		Average	
Temperature = 0		SpeedUp	τ	SpeedUp	τ	SpeedUp	τ	SpeedUp	τ	SpeedUp	τ
Q3 8B	PARD	5.02	6.33	4.91	5.97	4.96	6.28	2.68	3.28	4.39	5.46
	PARD-2	**5.71**	**7.38**	**5.20**	**6.56**	**5.63**	**7.23**	**2.74**	**3.35**	**4.82**	**6.13**
Q3 14B	PARD	5.16	6.07	5.10	5.88	4.98	5.82	2.78	3.21	4.51	5.25
	PARD-2	**5.63**	**7.16**	**5.28**	**6.61**	**5.50**	**6.97**	**2.76**	**3.14**	**4.79**	**5.97**
Q3 32B	PARD	5.18	6.46	4.90	5.99	4.76	5.83	2.64	2.99	4.37	5.32
	PARD-2	**5.71**	**7.12**	**5.10**	**6.24**	**5.25**	**6.47**	**2.67**	**3.02**	**4.68**	**5.71**

series, EAGLE-3 uses third-party trained weights (Tencent, 2025), while all other baselines and our method use official weights. For each method, we sweep the inference draft length and report the configuration yielding the highest speedup, with τ measured at the same draft length.

Target-Dependent Mode. Table 1 reports the main results under greedy decoding. Across all evaluated target models, PARD-2 consistently outperforms auto-regressive decoding and strong speculative decoding baselines in both speedup and average acceptance length τ. On Qwen3-8B, PARD-2 raises the average speedup to 5.81×, compared with 4.39× for PARD and 4.61× for DFlash, while increasing the average acceptance length to 7.38. Similar gains are observed on Qwen3-14B and Llama3.1-8B, where PARD-2 achieves 5.81× and 5.19× average speedups, respectively. It is worth noting that PARD-2 maintains strong performance on MT-Bench, which involves more complex multi-turn dialogue generation, suggesting that its benefits generalize beyond structured reasoning and coding benchmarks. These results demonstrate that PARD-2 improves the acceptance of consecutive draft tokens and translates this improvement into lossless inference acceleration in practice.

Target-Independent Mode. Table 2 evaluates PARD-2 in target-independent mode, where a single draft model serves Qwen3 targets from 8B to 32B without access to any target-side features. Even without target hidden states, PARD-2 delivers strong acceleration: on Qwen3-8B, it achieves an average speedup of 4.82× and acceptance length of 6.13, surpassing the target-dependent DFlash (4.61×, 5.69) on every benchmark and outperforming EAGLE-3 by 2.5×, despite both baselines relying on target features. Compared with PARD under the same setting, PARD-2 raises the average speedup by 6.2–9.8% and the mean acceptance length by 7.3–13.7%, with the largest gains on HumanEval and Math-500 (up to 13.7%). Moreover, PARD-2's average speedup remains stable across target scales (4.68–4.82×), and retains over 82% of its target-dependent speedup on Qwen3-8B and Qwen3-14B. These results show that PARD-2 provides a practical plug-and-play drafter: one model, trained once, accelerates an entire model family.

Large Batch Sizes Study. We further evaluate PARD-2 under large batch serving settings, where GPU utilization becomes increasingly important. Figure 1 reports both per-user throughput (TPS/User) and GPU throughput (TPS/GPU) across different batch sizes. PARD-2 consistently shifts the throughput frontier upward and to the right, indicating that it improves aggregate serving efficiency while maintaining higher per-user generation speed. Notably, even at batch size 64, where the speedup gain is relatively smaller due to higher GPU utilization, PARD-2 still outper-

Table 3: Effect of training data on Qwen3-8B. PARD is retrained on the same expanded data mixture as PARD-2 to disentangle the contribution of training data from that of the proposed method. All methods are evaluated with a fixed inference draft length of 16.

Method	Data	HumanEval		GSM8K		Math-500		MT-Bench		Average	
		Speedup	τ	Speedup	τ	Speedup	τ	Speedup	τ	Speedup	τ
PARD	Original	5.02	6.34	4.86	6.01	4.87	6.12	2.65	3.29	4.35	5.44
PARD	Expanded	5.20	6.60	5.10	6.28	5.34	6.70	2.80	3.44	4.61	5.75
PARD-2	Expanded	**6.34**	**8.09**	**6.20**	**7.68**	**6.56**	**8.27**	**3.15**	**3.89**	**5.56**	**6.98**

Table 4: Ablation study of core components and configurations in PARD-2. Starting from PARD, we progressively add target hidden features, CAT optimization, and multi-layer target features, and evaluate different draft lengths for training k_{train} and inference k_{infer}.

Method	Draft Length		HumanEval		GSM8K		MT-Bench		Average	
	k_{train}	k_{infer}	Speedup	τ	Speedup	τ	Speedup	τ	Speedup	τ
PARD	8	8	4.71	5.90	3.89	4.78	2.75	3.42	3.78	4.70
PARD + Target Feat	8	8	4.92	6.18	4.18	5.12	2.88	3.57	3.99	4.96
PARD + Target Feat	16	8	4.76	6.00	4.11	5.02	2.80	3.46	3.89	4.83
PARD + Target Feat + CAT	16	8	4.94	6.19	4.24	5.15	2.93	3.60	4.04	4.98
PARD + Target Feat + CAT	16	16	5.86	7.90	4.41	5.79	2.79	3.67	4.35	5.79
PARD + Multi-layer Target Feat + CAT	16	16	5.89	7.93	4.41	5.80	2.86	3.79	4.39	5.84

forms PARD on both Llama3.1-8B and Qwen3-8B. These results show that the gains of PARD-2 are not limited to small-batch; instead, PARD-2 remains effective in high-throughput serving scenarios, where large-batch decoding is commonly used to maximize GPU utilization.

4.3 ABLATION STUDIES

In this section, we ablate the key design choices of PARD-2, including the effect of training data, the effectiveness of CAT, the impact of stochastic gating for target features, and a breakdown of the improvements over PARD. Unless otherwise specified, all ablation models are trained for 30k steps on Llama3.1-8B and evaluated with vLLM under the same setting as the main experiments.

Effect of Training Data. Since PARD-2 is trained on an expanded data mixture, we retrain PARD on exactly the same mixture to disentangle the contribution of training data from that of our method. As shown in Table 3, the expanded data improves PARD's τ by 0.15–0.58 (+4.1–9.5%), whereas moving from PARD to PARD-2 under the same data yields a further gain of 0.45–1.57 (+13.1–23.4%). Consequently, the data mixture accounts for only 15–27% of the total improvement in τ, while the remaining 73–85% stems from CAT and the target-aligned design of PARD-2.

Confidence-Adaptive Token Optimization (CAT). CAT reweights the token-level training loss according to the target model's confidence, prioritizing tokens that are likely to extend the accepted prefix rather than supervising all positions uniformly. As shown in Table 4, naively increasing the training draft length k_{train} from 8 to 16 degrades τ from 4.96 to 4.83, as hard-to-predict distant tokens dominate the uniform loss. CAT resolves this issue: under the same configuration, it raises τ to 4.98 and the average speedup from $3.89\times$ to $4.04\times$, with consistent gains on all three benchmarks. More importantly, by making long-horizon training effective, CAT enables longer inference drafts: increasing k_{infer} to 16 further lifts τ to 5.79 and the average speedup to $4.35\times$.

To further validate the design of CAT, we compare it against a fixed position-wise decay strategy (Chen et al., 2026; Liu et al., 2026) ($\gamma_t = \gamma^{t-1}$), a common heuristic in parallel drafting, where $\gamma = 1.0$ corresponds to uniform supervision. As reported in Table 5, position-wise decay yields modest gains at $\gamma = 0.8$ (τ from 5.52 to 5.61), but is highly sensitive to the decay rate: a more aggressive $\gamma = 0.6$ reduces τ to 5.12, falling below uniform supervision. In contrast, CAT requires no decay-rate tuning and outperforms the best decay setting on all three benchmarks, raising the average τ to 5.79 and the average speedup from $4.23\times$ to $4.36\times$. These results indicate that adapting the weights to the confidence of each token and its prefix is effective for parallel draft training.

Stochastic Gating for Target Features. To balance target-dependent performance and target-independent versatility, we introduce a training-time stochastic gate for target-feature injection. The gate disables target features with probability ρ and injects them otherwise, preventing the drafter

Table 5: Comparison between fixed-decay and token-adaptive weighting strategies.

Prob Loss	HumanEval		GSM8K		MT-Bench		Average	
	Speedup	τ	Speedup	τ	Speedup	τ	Speedup	τ
$\gamma = 1.0$	5.52	7.45	4.22	5.58	2.67	3.54	4.14	5.52
$\gamma = 0.8$	5.61	7.56	4.33	5.67	2.74	3.60	4.23	5.61
$\gamma = 0.6$	4.89	6.58	3.99	5.21	2.72	3.58	3.87	5.12
Ours	**5.86**	**7.90**	**4.43**	**5.79**	**2.79**	**3.67**	**4.36**	**5.79**

Table 6: Effect of the stochastic gating ratio.ρ = 0.1 achieves the best trade-off.

Gate Ratio	HumanEval		GSM8K		MT-Bench		Average	
	Speedup	τ	Speedup	τ	Speedup	τ	Speedup	τ
w/o gate	6.10	8.28	4.66	6.17	2.88	3.86	4.55	6.10
$\rho = 0.5$	5.96	8.15	4.55	6.01	2.85	3.80	4.45	5.99
$\rho = 0.2$	5.99	8.18	4.66	6.12	2.85	3.80	4.50	6.03
$\rho = 0.1$	**6.02**	**8.19**	**4.68**	**6.13**	**2.88**	**3.85**	**4.53**	**6.06**

from over-relying on target hidden states and enabling the same model to operate without them at inference time. As shown in Table 6, gating with $\rho = 0.1$ incurs negligible cost in target-dependent mode, reducing τ from 6.10 to 6.06 and the average speedup from $4.55\times$ to $4.53\times$ (less than 1%), whereas a larger $\rho = 0.5$ leads to a more noticeable drop ($\tau = 5.99$). In exchange, this single model also serves as a strong target-independent drafter (Table 2), making $\rho = 0.1$ a favorable trade-off.

Analysis of Performance Gains. Table 4 ablates the key components of PARD-2. Feeding target hidden states to the draft model as conditioning features raises the average τ from 4.70 to 4.96, with the largest gain on GSM8K. Naively extending the training draft length k_{train} to 16 degrades τ to 4.83, as distant tokens dominate the uniform loss; CAT recovers this loss and enables effective long-horizon training (detailed above). Building on this, increasing the inference draft length k_{infer} to 16 yields the largest single gain, lifting τ to 5.79, and multi-layer target features further improve it to 5.84. Overall, PARD-2 improves the average τ by 24.3% and the speedup by 16.1% over PARD.

5 Related Work

Speculative decoding (Leviathan et al., 2023; Chen et al., 2023) alleviates the memory-bandwidth bottleneck in auto-regressive generation by using a lightweight draft model to propose tokens for parallel verification by a target LLM. To improve draft-target alignment, Medusa (Cai et al., 2024), GLIDE and CAPE (Du et al., 2024), and the EAGLE series (Li et al., 2024a; 2025b) incorporate the KV cache or hidden features of the target model, while DistillSpec (Zhou et al., 2023) employs knowledge distillation. To further minimize wall-clock latency, methods such as PEARL (Liu et al., 2025) and SSD (Kumar et al., 2026) decouple drafting and verification for parallel execution, whereas SpecInfer (Miao et al., 2023), Falcon (Gao et al., 2024) and EAGLE-2 (Li et al., 2024b) introduce advanced tree-based verification. Training-free n-gram matching methods such as LOOKAHEAD (Fu et al., 2024) and PROMTEC (Lee et al., 2025) also accelerate inference.

Despite these advances, many SD methods still rely on auto-regressive drafting, whose sequential dependency limits drafting throughput. Recent parallel drafting methods address this limitation by predicting multiple future tokens in a single forward pass. ParallelSpec (Xiao et al., 2024) trains a parallel drafter that generates multiple tokens in a single forward pass. P-EAGLE (Hui et al., 2026) and PARD (An et al., 2025) adapt auto-regressive models to parallel masked prediction, while SpecDiff (Christopher et al., 2025), SpecDiff-2 (Sandler et al., 2025), DART (Liu et al., 2026) and DFlash (Chen et al., 2026) employ diffusion-style drafters for parallel token generation. However, existing parallel methods often rely on uniform token-level supervision. While some approaches (Chen et al., 2026; Liu et al., 2026) introduce fixed position-aware decaying weights, these weights depend solely on position and ignore the prefix context. In practice, whether a token is accepted depends not only on its own prediction quality but also on whether its entire prefix is accepted, suggesting that supervision weights should account for both factors.

6 Conclusion

We propose PARD-2, a dual-mode speculative decoding framework that unifies target-dependent and target-independent drafting within a single draft model. By analyzing the expected acceptance length, we identify a gap between per-token training objectives and the inference-time goal of maximizing consecutive token acceptance. To bridge this gap, we introduce Confidence-Adaptive Token (CAT) optimization, which reweights each token by the cumulative target-model confidence over its prefix. Experiments across models and benchmarks show that PARD-2 consistently outperforms existing SD methods, and even in target-independent mode surpasses target-dependent baselines.

AI USE STATEMENT

In this work, we used generative AI tools to assist with a small amount of auxiliary infrastructure code (e.g., scripts for launching inference servers) and for language editing to improve readability. We have not used generative AI tools for formulating research ideas, designing or implementing the proposed method, designing experiments, generating data, or interpreting results. All AI-assisted code was reviewed and tested by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the details necessary to reproduce our results. The CAT objective, the full training loss, stochastic gating, and dual-mode inference are described in Section 3. Section 4 specifies the training data, target models, hyperparameters (ρ, β, and draft length), hardware, and evaluation settings, including the vLLM configuration and the draft-length selection protocol. All baselines use publicly available weights. We will publicly release our code and model checkpoints.

REFERENCES

Zihao An, Huajun Bai, Ziqiong Liu, Dong Li, and Emad Barsoum. Pard: Accelerating llm inference with low-cost parallel draft model adaptation. *arXiv preprint arXiv:2504.18583*, 2025.

Zachary Ankner, Rishab Parthasarathy, Aniruddha Nrusimha, Christopher Rinard, Jonathan Ragan-Kelley, and William Brandon. Hydra: Sequentially-dependent draft heads for medusa decoding. *arXiv preprint arXiv:2402.05109*, 2024.

Aarti Basant, Abhijit Khairnar, Abhijit Paithankar, Abhinav Khattar, Adithya Renduchintala, Aditya Malte, Akhiad Bercovich, Akshay Hazare, Alejandra Rico, Aleksander Ficek, et al. Nvidia nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model. *arXiv preprint arXiv:2508.14444*, 2025.

Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, et al. Nemotron 3 nano: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning. *arXiv preprint arXiv:2512.20848*, 2025.

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

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

Jian Chen, Yesheng Liang, and Zhijian Liu. Dflash: Block diffusion for flash speculative decoding. *arXiv preprint arXiv:2602.06036*, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Jacob K Christopher, Brian R Bartoldson, Tal Ben-Nun, Michael Cardei, Bhavya Kailkhura, and Ferdinando Fioretto. Speculative diffusion decoding: Accelerating language generation through diffusion. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 12042–12059, 2025.

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

Cunxiao Du, Jing Jiang, Xu Yuanchen, Jiawei Wu, Sicheng Yu, Yongqi Li, Shenggui Li, Kai Xu, Liqiang Nie, Zhaopeng Tu, et al. Glide with a cape: A low-hassle method to accelerate speculative decoding. *arXiv preprint arXiv:2402.02082*, 2024.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. Break the sequential dependency of llm inference using lookahead decoding. *arXiv preprint arXiv:2308.16710*, 2024. URL https://arxiv.org/abs/2308.16710.

Xiangxiang Gao, Weisheng Xie, Yiwei Xiang, and Feng Ji. Falcon: Faster and parallel inference of large language models through enhanced semi-autoregressive drafting and custom-designed decoding tree. *arXiv preprint arXiv:2412.12639*, 2024.

Mude Hui, Xin Huang, Jaime Campos Salas, Yue Sun, Nathan Pemberton, Xiang Song, Ashish Khetan, and George Karypis. P-eagle: Parallel-drafting eagle with scalable training. *arXiv preprint arXiv:2602.01469*, 2026.

Tanishq Kumar, Tri Dao, and Avner May. Speculative speculative decoding. In *The Fourteenth International Conference on Learning Representations*, 2026.

Alan Chi-Man Lee, Wing-Sun Cheng, and Calvin Chun-Kit Chan. Promtec: Fast llm inference decoding using prompt multi-lookup with template database and common sequences. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6830–6842, 2025.

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

Guanghao Li, Zhihui Fu, Min Fang, Qibin Zhao, Ming Tang, Chun Yuan, and Jun Wang. Diffuspec: Unlocking diffusion language models for speculative decoding. *arXiv preprint arXiv:2510.02358*, 2025a.

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

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-2: Faster inference of language models with dynamic draft trees, 2024b. URL https://arxiv.org/abs/2406.16858.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *arXiv preprint arXiv:2503.01840*, 2025b.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Fuliang Liu, Xue Li, Ketai Zhao, Yinxi Gao, Ziyan Zhou, Zhonghui Zhang, Zhibin Wang, Wanchun Dou, Sheng Zhong, and Chen Tian. Dart: Diffusion-inspired speculative decoding for fast llm inference. *arXiv preprint arXiv:2601.19278*, 2026.

Tianyu Liu, Yun Li, Qitan Lv, Kai Liu, Jianchen Zhu, Winston Hu, and Xiao Sun. Pearl: Parallel speculative decoding with adaptive draft length. In *The Thirteenth International Conference on Learning Representations*, 2025.

AI @ Meta Llama Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct, 2023.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, et al. Specinfer: Accelerating generative large language model serving with tree-based speculative inference and verification. *arXiv preprint arXiv:2305.09781*, 2023.

Jameson Sandler, Jacob K Christopher, Thomas Hartvigsen, and Ferdinando Fioretto. Specdiff-2: Scaling diffusion drafter alignment for faster speculative decoding. *arXiv preprint arXiv:2511.00606*, 2025.

Tencent. AngelSlim. `https://github.com/Tencent/AngelSlim`, June 2025. GitHub repository.

Zilin Xiao, Hongming Zhang, Tao Ge, Siru Ouyang, Vicente Ordonez, and Dong Yu. Parallelspec: Parallel drafter for efficient speculative decoding. *arXiv preprint arXiv:2410.05589*, 2024.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*, 2024.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. Distillspec: Improving speculative decoding via knowledge distillation. *arXiv preprint arXiv:2310.08461*, 2023.

A APPENDIX

A.1 TRAINING HYPERPARAMETERS

Table 7 summarizes the hyperparameters used for training.

Table 7: Selected Hyperparameters for PARD-2 Training

Hyperparameter	**Llama3.1-8B**	**Qwen3-8B**	**Qwen3-14B**
Optimizers	AdamW	AdamW	AdamW
Learning Rate	1e-5	3e-5	3e-5
Per Device Train Batch Size	8	4	4
Gradient Accumulation Steps	1	2	2
Num Processes	8	8	8
Num Train Epochs	4	2	2
Training Draft Length K	16	16	16
Stochastic Gating Ratio ρ	0.1	0.1	0.1
CE Loss Coefficient β	0.1	0.1	0.1
Max Seq Length	512	1024	1024

A.2 LIMITATIONS

While PARD-2 performs well in small-batch and high-concurrency settings, our current implementation does not use tree attention. Instead, we adopt single-chain (non-tree) drafting for efficiency and simplicity. This design choice may limit the potential performance of PARD-2 in small-batch scenarios, where tree attention could better exploit multiple candidate branches and improve acceptance. However, in large-batch settings, tree attention often introduces substantial computational and memory overhead, which can offset its benefits; therefore, this limitation has little impact on our main high-concurrency use cases. Exploring how to efficiently support tree attention under large-batch serving remains an interesting direction for future work.