BEYOND THE POINT MASS: ACCEPTANCE-DERIVED WEIGHTS AND DISTRIBUTIONAL LOSSES FOR SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

In speculative decoding, a lightweight draft model proposes tokens that the target model verifies in parallel, speeding up generation without changing the target distribution. D-PACE (Wu et al., 2026) trains such drafters with a loss that weights each position by its contribution to the expected accepted length. Both its weights and its cross-entropy (CE) loss use only the draft's probability of the recorded target token, treating the target as a point mass. Under rejection sampling, however, a sampled draft token is accepted with probability equal to the overlap of the draft and target distributions, one minus their total variation (TV). We derive *D-PAL (Dynamic Position-wise Acceptance Loss)* from the exact expected accepted length, using this overlap as the acceptance signal. This yields overlap-based position weights with log acceptance (LA) as the per-position loss, and the objective reduces to D-PACE for a point-mass target. Since the weights are detached, any per-position loss can replace LA in practice. We show that forward KL divergence and Rényi divergence of order $1/2$ match or exceed LA, and that Rényi-$1/2$ provably bounds the true acceptance rate from above and below. Replacing CE with Rényi-$1/2$ improves acceptance length under both D-PACE's weights and ours. On six benchmarks with a Qwen3-4B target, D-PAL raises mean acceptance length over D-PACE from 4.50 to 4.57 at $T = 0$ and from 4.20 to 4.28 at $T = 1$, and consistently achieves higher throughput at both temperatures.

1 INTRODUCTION

Speculative decoding accelerates autoregressive generation without changing the target model's output distribution (Leviathan et al., 2023; Chen et al., 2023): a lightweight draft model proposes tokens, and the target model verifies them in one parallel pass. DFlash drafts a whole block in one forward pass (Chen et al., 2026). Since verification stops at the first rejection, early block positions matter most. DFlash trains with cross-entropy (CE) and a fixed position decay, and DSpark and H-Spec add a total variation (TV) term to CE under the same decay (Cheng et al., 2026; Jiang et al., 2026). LK Losses optimizes acceptance directly, with the negative log acceptance or an adaptive mix of Kullback–Leibler (KL) divergence and TV, but still uses fixed decay (Samarin et al., 2026). D-PACE (Dynamic Position-Aware Cross-Entropy) instead derives position weights from a surrogate of the expected accepted length (Wu et al., 2026). SpecForge (SpecForge Contributors, 2026), Speculators (Speculators Contributors, 2026), and TorchSpec (TorchSpec Contributors, 2026) use it to train DFlash drafters.

D-PACE treats the target as a point mass on the recorded token $y_t^\star$: it estimates acceptance by the draft's confidence $q_t(y_t^\star)$ and fits that token with CE. Under standard rejection sampling, however, acceptance depends on the full target distribution p_t through the overlap $\widehat{\alpha}_t = \sum_v \min(p_{t,v}, q_{t,v}) = 1 - \mathrm{TV}(p_t, q_t)$ (Leviathan et al., 2023). When the target is uncertain, confidence can underestimate this overlap by a wide margin. For example, if $p_t = q_t$ is uniform over k tokens, the verifier always accepts a sampled draft token ($\widehat{\alpha}_t = 1$), yet $q_t(y_t^\star) = 1/k$. CE likewise fits only the recorded token.

We propose **D-PAL** (**D**ynamic **P**osition-wise **A**cceptance **L**oss), which we derive by differentiating the expected accepted length with the *true* acceptance signal $1 - \mathrm{TV}(p_t, q_t)$. The derivation yields overlap-based position weights paired with a log acceptance (LA) loss, $-\log \widehat{\alpha}_t$, as the local loss,

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.

which reduces to D-PACE when p_t is a point mass. Because D-PAL detaches the weights, we can swap LA for any local loss in practice. We write D-PAL[X] for D-PAL with local loss X. We show overlap bounds the acceptance of the argmax proposals that deployed DFlash uses (Proposition 1). Replacing CE with a distributional loss, Rényi divergence of order $1/2$, raises mean acceptance length under both confidence and overlap weights. In our experiments, we show that LA is not the best local loss. It is not differentiable where a draft probability equals the target probability, and its gradient with respect to the draft logits does not shrink near $q_t = p_t$, whereas the KL and Rényi gradients vanish there. Forward KL and Rényi-$1/2$ match or exceed LA, and Rényi-$1/2$, unlike KL, bounds overlap from both sides (Lemma 1). We also test TV-based losses, where pure TV collapses and CE+TV falls short, and analyze their gradients.

Our contributions are:

i) **Derivation from expected accepted length.** Differentiating the exact expected accepted length with the overlap as the acceptance signal gives overlap-based weights paired with LA. Proposition 1 connects the overlap to the argmax verification used in deployment. We show D-PAL[LA] improves on D-PACE, raising mean acceptance length on Qwen3-4B from 4.50 to 4.54 at $T = 0$ and from 4.20 to 4.24 at $T = 1$ (Table 1).

ii) **Local loss beyond log acceptance.** The derivation points to LA, but LA is not differentiable where the draft matches the target. Keeping the same weights, we replace it with forward KL and with Rényi-$1/2$, and both reach the same or higher acceptance length (Table 1). We also show Rényi-$1/2$ bounds overlap from above and below (Lemma 1).

iii) **Consistent gains over D-PACE and DFlash.** In experiments across temperatures, block sizes, draft depths, target models at the 4B and 8B scale, and thinking modes, we show that **D-PAL consistently achieves longer acceptance lengths and higher throughputs than both D-PACE (up to 5%) and DFlash (up to 11%).**

2 Background

2.1 Speculative decoding with DFlash

In speculative decoding (Leviathan et al., 2023; Chen et al., 2023), the target model scores a block of B draft proposals in one forward pass and accepts a contiguous prefix of length τ.

DFlash (Chen et al., 2026) drafts with a small block-diffusion model that projects hidden states from a few target layers into the keys and values of every draft layer. A block of B positions starts with an anchor, the last verified token, followed by $D = B - 1$ mask tokens. In one forward pass, the drafter predicts distributions $q_1, \dots, q_D$ for all masked positions. Each q_t thus depends on the prefix but not on earlier proposals in the block, while the target's p_t conditions on all tokens before t.

In training, DFlash samples anchor positions in responses generated by the frozen target and predicts the next D recorded tokens $y_1^\star, \dots, y_D^\star$ from each anchor context. Since a proposal counts only if every earlier one is accepted, DFlash fits $y_t^\star$ with CE and weights positions by a fixed exponential decay,

$$\mathcal{L}_{\text{DFlash}} = \sum_{t=1}^{D} e^{-(t-1)/\gamma} \left[-\log q_t(y_t^\star) \right], \tag{1}$$

with $\gamma = 7$ for $B = 16$. The frozen target output head also gives the full target distribution p_t at each position, but DFlash does not use it.

2.2 Expected accepted length

Let $L = \tau - 1$ count the accepted draft tokens, and let $\alpha_t = \Pr(L \geq t \mid L \geq t - 1)$ be the conditional acceptance probability at position t. The tail-sum identity gives the exact expected accepted length (Li et al., 2026b; Wu et al., 2026)

$$\mathbb{E}[L] = \sum_{k=1}^{D} \prod_{i=1}^{k} \alpha_i. \tag{2}$$

Each α_t appears in every accepted prefix ending at $k \geq t$. For positive α_t, its relative contribution and the gradient of $\mathbb{E}[L]$ are

$$w_t^{\text{raw}} = \alpha_t \frac{\partial \mathbb{E}[L]}{\partial \alpha_t} = \sum_{k=t}^{D} \prod_{i=1}^{k} \alpha_i, \qquad \nabla \mathbb{E}[L] = \sum_{t=1}^{D} w_t^{\text{raw}} \nabla \log \alpha_t. \tag{3}$$

Every term is positive, so w_t^{raw} decreases with t.

2.3 Position weights and local losses

Equation (3) suggests drafter objectives for a block of the form $\sum_{t=1}^{D} w_t \ell_t$, with weights computed on target-generated training contexts. DFlash (Chen et al., 2026) uses a fixed exponential decay $e^{-(t-1)/\gamma}$ to emphasize early positions (Equation (1)). D-PACE (Wu et al., 2026) instead uses the draft's confidence $q_t(y_t^\star)$ as a proxy for α_t, smooths it to $\tilde{q}_t = \rho + (1-\rho) q_t(y_t^\star)$ with $\rho \in (0,1)$, and builds detached weights paired with CE:

$$w_t^{\text{PACE}} = \sum_{k=t}^{D} \prod_{i=1}^{k} \tilde{q}_i, \qquad \mathcal{L}_{\text{D-PACE}} = \sum_{t=1}^{D} w_t^{\text{PACE}} [-\log q_t(y_t^\star)]. \tag{4}$$

With $\rho > 0$, an early low-confidence position cannot zero out all later weights.

3 D-PAL: Acceptance in the Weights, Fit in the Loss

3.1 Acceptance-derived weights from overlap

D-PACE's unsmoothed confidence is overlap with a point mass on the recorded target token: $\sum_v \min\{\delta_{y^\star,v}, q_v\} = q(y^\star)$. Standard rejection sampling instead compares the draft with the full target distribution. At a fixed context with target and draft distributions p_t, q_t, the verifier accepts $y \sim q_t$ with probability $\min\{1, p_t(y)/q_t(y)\}$. Averaging over y gives the one-token acceptance (Leviathan et al., 2023; Chen et al., 2023)

$$\widehat{\alpha}_t = \sum_v \min\{p_{t,v}, q_{t,v}\} = 1 - \text{TV}(p_t, q_t), \qquad \text{TV}(p,q) := \tfrac{1}{2}\|p - q\|_1. \tag{5}$$

D-PAL applies the weighting rule of Equation (3) to this overlap. On target-generated sequences, the product of overlaps through position k approximates the chance that sampled proposals survive to k. As in D-PACE, we smooth only inside the products, $s_t = \rho + (1-\rho)\widehat{\alpha}_t$, recompute the weights from the current draft, and detach them:

$$w_t = \sum_{k=t}^{D} \prod_{i=1}^{k} s_i, \qquad \mathcal{L}_{\text{D-PAL}} = \sum_{t=1}^{D} w_t \, \ell_t, \tag{6}$$

where ℓ_t is the local loss chosen in Section 3.2. Equivalently,

$$w_t = \Big(\prod_{i \leq t} s_i\Big)\Big(1 + \sum_{k>t} \prod_{t<i\leq k} s_i\Big), \tag{7}$$

the smoothed acceptance through t times the continuation value after t (Figure 1).

The SGLang DFlash backend proposes the draft argmax $\hat{y}_t = \arg\max_v q_{t,v}$ at both temperatures, so proposals are never sampled from q_t. Overlap still bounds the acceptance of this proposal.

Proposition 1. *(i) Let Δ_t be the gap between the largest and second-largest target probabilities. If $\text{TV}(p_t, q_t) < \Delta_t/2$, then $\hat{y}_t = \arg\max_v p_{t,v}$. (ii) $p_t(\hat{y}_t) \geq \max_v q_{t,v} - \text{TV}(p_t, q_t)$.*

At $T = 0$, the verifier accepts $\hat{y}_t$ if it matches the target argmax, which (i) guarantees whenever the target margin exceeds $2\,\text{TV}$. At $T = 1$, SGLang's target-only verification accepts it with probability $p_t(\hat{y}_t)$ (SGLang Contributors, 2026), which (ii) bounds from below (proof in Appendix B.2).

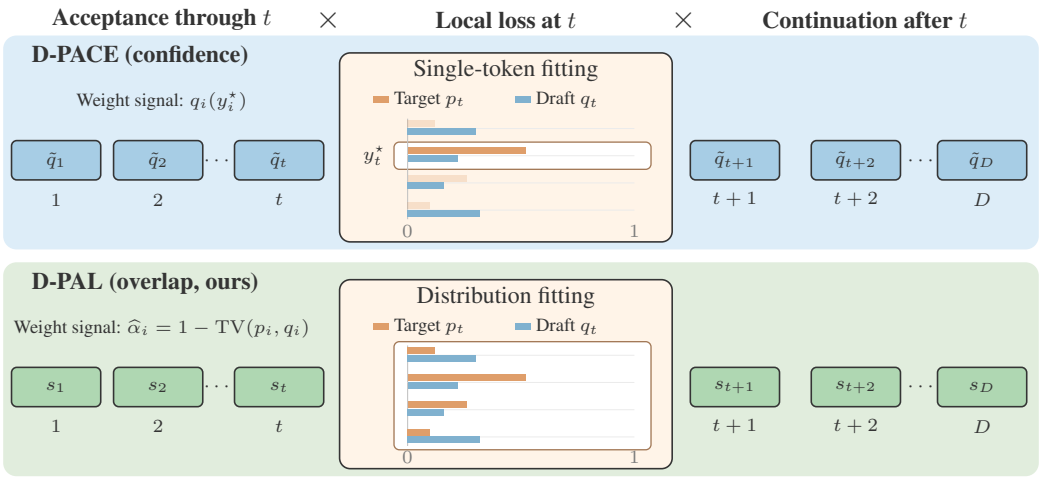


Figure 1: **D-PAL uses the full target distribution for position weighting and, with KL or Rényi, for local fitting.** Top: D-PACE weights by confidence and fits one token with CE; faded bars are target mass CE ignores. Bottom: D-PAL weights by overlap and, with a distributional loss such as KL or Rényi, fits q_t to all of p_t. Each weighted term $w_t \ell_t$ is the product of the acceptance through t (left), the local loss at t (center), and the continuation value after t (right).

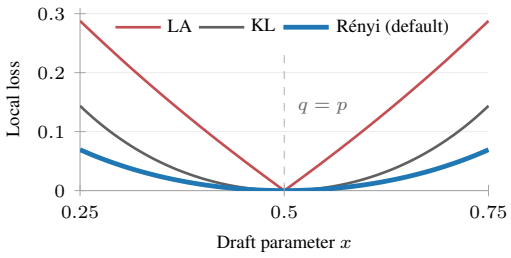


Figure 2: **Near $q = p$, LA's slope stays near ± 1 and flips sign, while KL and Rényi-$1/2$ flatten out.** Binary example with target $p = (0.5, 0.5)$ and draft $q = (x, 1 - x)$; slopes are with respect to x, and all three losses are zero at $x = 0.5$.

3.2 Log acceptance loss and beyond

D-PAL[LA]. Equation (3) pairs each weight with $\nabla \log \alpha_t$, so gradient ascent on $\mathbb{E}[L]$ matches gradient descent on the log-acceptance (LA) loss $\ell_t = -\log \widehat{\alpha}_t$, which gives D-PAL[LA]. For a point-mass target, $\widehat{\alpha}_t = q_t(y_t^\star)$, so LA becomes CE, as LK Losses notes (Samarin et al., 2026), and D-PAL[LA] reduces to D-PACE. At $\rho = 0$, gradient descent on D-PAL[LA] follows gradient ascent on the overlap-based prefix-sum surrogate (Appendix B.3); as in D-PACE, we keep the floor $\rho > 0$ as a regularizer.

However, LA is not differentiable where a draft probability equals the corresponding target probability. Its gradient is the TV gradient scaled by $1/\widehat{\alpha}_t$. With respect to the draft logits, the TV gradient is $\frac{1}{2}\, q_t \odot (\sigma_t - \mathbb{E}_{q_t}[\sigma_t])$ with $\sigma_{t,v} = \mathrm{sign}(q_{t,v} - p_{t,v})$ (Samarin et al., 2026). It does not vanish as q_t approaches p_t, and changes sign where $q_{t,v}$ crosses $p_{t,v}$. On the other hand, KL and Rényi logit gradients vanish at $q_t = p_t$ (Appendix B.1). Figure 2 shows a binary example. Because w_t is detached, any local loss can replace LA without changing the weight formula; in the following, we consider D-PAL[KL] and D-PAL[Rényi] as smooth alternatives to D-PAL[LA].

D-PAL[KL]. Cross-entropy, the local loss of D-PACE, is smooth but fits one sampled token only. If $y \sim p_t$, then $\mathbb{E}_y[-\log q_t(y)] = H(p_t) + \mathrm{KL}(p_t \| q_t)$, and the logit gradient $q_t - e_y$ has mean $q_t - p_t$. Up to the constant $H(p_t)$, CE is thus a one-sample estimate of forward KL, with label noise at every position. Our labels are sampled at positive temperature, so CE also fits a sharper target than the $T = 1$ distribution p_t that sets acceptance. D-PAL[KL] fits this forward KL directly instead.

D-PAL[Rényi]. KL fits the full distribution, but it bounds overlap only from below and can be arbitrarily large while overlap is close to one. D-PAL[Rényi] instead replaces the minimum in

overlap with a geometric mean, the Bhattacharyya coefficient $\mathrm{BC}(p, q) = \sum_v \sqrt{p_v q_v}$, whose scaled negative logarithm is the Rényi divergence of order $1/2$ (Rényi, 1961):

$$D_{1/2}(p\|q) = -2 \log \mathrm{BC}(p, q), \tag{8}$$

$$\mathcal{L}_{\text{D-PAL[Rényi]}} = \sum_{t=1}^{D} w_t D_{1/2}(p_t\|q_t). \tag{9}$$

Its logit gradient is $q_t - r_t$ with $r_{t,v} \propto \sqrt{p_{t,v} q_{t,v}}$ (Appendix B.1), which varies continuously with the target–draft mismatch. The Bhattacharyya coefficient bounds overlap from above and below.

Lemma 1 (Overlap bounds). *For categorical distributions p, q, $\alpha(p, q) = 1 - \mathrm{TV}(p, q)$ satisfies*

$$1 - \sqrt{1 - e^{-D_{1/2}(p\|q)}} \;\le\; \alpha(p, q) \;\le\; e^{-D_{1/2}(p\|q)/2}. \tag{10}$$

This bound is the classical inequality between TV and the Bhattacharyya coefficient (Fuchs & van de Graaf, 1999) (proof in Appendix B.2). Taking negative logarithms gives $D_{1/2}/2 \le -\log \widehat{\alpha}_t \le -\log\big(1 - \sqrt{1 - e^{-D_{1/2}}}\big)$. D-PAL[Rényi] minimizes the smooth quantity $D_{1/2}$ that controls both bounds, so driving it to zero drives LA to zero. This lower bound on overlap is never below the Bretagnolle–Huber bound from KL (van Erven & Harremoës, 2014; Canonne, 2023) (Appendix B.1).

3.3 D-PAL Algorithm

Algorithm 1 shows how to compute the D-PAL loss. Any of the previously discussed losses (LA, KL, Rényi, etc.) can be plugged in for l_t.

Algorithm 1: D-PAL loss computation

Input: target p, draft q, valid mask m, smoothing ρ, and chosen local loss ℓ_t.

for each nonempty valid block b:
  Select proposals with $m_{b,t} = 1$ in order; index them by $t = 1, \ldots, D_b$.
  $\widehat{\alpha}_t \leftarrow \sum_v \min(p_{t,v}, q_{t,v})$; $s_t \leftarrow \rho + (1 - \rho)\widehat{\alpha}_t$ **for all** t
  $\widetilde{R}_0 \leftarrow 1$; $w_{D_b+1} \leftarrow 0$
  for $t = 1, \ldots, D_b$: $\widetilde{R}_t \leftarrow s_t \widetilde{R}_{t-1}$ ▷ *Smoothed prefix product*
  for $t = D_b, \ldots, 1$: $w_t \leftarrow w_{t+1} + \widetilde{R}_t$ ▷ *Position weights*
  Detach $w_{1:D_b}$.
  $\mathcal{L}_b \leftarrow \sum_{t=1}^{D_b} w_t \ell_t$
return Mean of $\mathcal{L}_b$ within each sequence, then across sequences.

4 Experiments

4.1 Experimental setup

Models and training. We train DFlash drafters in SpecForge with frozen Qwen3-4B, Qwen3-8B, and Llama-3.1-8B-Instruct targets (Chen et al., 2026; Li et al., 2026a; Yang et al., 2025; Grattafiori et al., 2024). Within each comparison, methods share target-generated data, optimizer, schedule, and decoding (Appendix A). Table 1 uses the Qwen3-4B-Instruct-100K dataset released by the DFlash authors. Following D-PACE, we use three-layer drafts and block size $B = 16$ (Chen et al., 2026; Wu et al., 2026). Request grouping uses these same checkpoints. For the ablations and other targets, we use SpecForge to sample a 36K-example ShareGPT subset; ablations use Qwen3-4B and $B = 16$ unless varied.

Evaluation. We evaluate on GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024) for math, HumanEval and MBPP (Chen et al., 2021; Austin et al., 2021) for code, and first-turn MT-Bench (Zheng et al., 2023) and Alpaca (Taori et al., 2023) for chat. We run the six-benchmark evaluations with SGLang DFlash on one NVIDIA L40S, one request at a time, at $T = 0$ and $T = 1$. Our two primary evaluation metrics are speedup ratio (SR) (throughput divided

Table 1: **D-PAL[Rényi], D-PAL[LA], and D-PAL[KL] beat D-PACE; D-PAL[TV] and D-PAL[CE+TV] fall short.** Qwen3-4B target, three-layer DFlash drafts, $B = 16$, 100K training examples, $\rho = 0.3$ for D-PAL and 0.5 for D-PACE. SR is speedup over autoregressive decoding, τ is acceptance length, and bold marks the best value.

	MATH				CODE				CHAT					
	GSM8K		MATH-500		HumanEval		MBPP		MT-Bench		Alpaca		**Mean**	
Method	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ	SR	τ
Baselines	*Temperature* $T = 0$													
DFlash	3.20	4.50	3.94	5.46	3.33	4.67	3.06	4.28	1.97	3.15	1.77	2.71	2.88	4.13
D-PACE	3.50	4.96	4.35	6.06	3.60	5.09	3.34	4.65	2.05	3.36	1.83	2.87	3.11	4.50
D-PAL weights														
D-PAL[LA]	3.50	5.03	4.32	6.06	3.65	5.15	**3.38**	4.70	2.04	3.41	1.84	2.90	3.12	4.54
D-PAL[KL]	3.50	5.06	**4.43**	**6.11**	**3.68**	**5.19**	**3.38**	**4.71**	**2.08**	3.42	**1.87**	**2.92**	**3.16**	**4.57**
D-PAL[Rényi]	**3.53**	**5.07**	4.39	6.10	3.66	5.17	3.31	4.68	**2.08**	**3.44**	1.86	**2.92**	3.14	4.56
D-PAL[TV]	0.75	1.01	0.76	1.02	0.77	1.04	0.77	1.03	0.77	1.05	0.77	1.07	0.77	1.04
D-PAL[CE+TV]	3.39	4.89	4.24	5.86	3.50	4.92	3.27	4.54	1.94	3.23	1.77	2.86	3.02	4.39
D-PAL[KL+TV]	3.49	5.03	4.33	6.05	3.64	5.14	3.33	4.68	2.04	3.40	1.85	2.88	3.11	4.53
Baselines	*Temperature* $T = 1$													
DFlash	2.99	4.29	3.25	4.80	3.07	4.40	2.85	4.09	1.84	3.04	1.71	2.71	2.62	3.89
D-PACE	3.24	4.76	3.54	5.31	3.30	4.75	3.03	4.41	1.90	3.18	1.75	2.82	2.79	4.20
D-PAL weights														
D-PAL[LA]	3.28	4.78	3.52	5.26	**3.37**	4.81	3.05	4.48	1.91	3.26	1.76	2.85	2.82	4.24
D-PAL[KL]	3.28	4.81	3.49	5.27	3.36	4.82	3.05	4.47	1.91	3.25	1.78	2.85	2.81	4.25
D-PAL[Rényi]	**3.34**	**4.84**	3.52	**5.35**	**3.37**	4.82	**3.10**	4.48	**1.94**	3.28	**1.81**	**2.89**	**2.85**	**4.28**
D-PAL[TV]	0.75	1.01	0.76	1.02	0.77	1.04	0.77	1.03	0.77	1.05	0.77	1.06	0.76	1.04
D-PAL[CE+TV]	3.18	4.64	3.48	5.19	3.23	4.61	2.96	4.32	1.85	3.11	1.69	2.78	2.73	4.11
D-PAL[KL+TV]	3.25	4.77	**3.58**	5.34	**3.37**	**4.88**	3.07	**4.49**	1.91	**3.30**	1.77	2.84	2.83	4.27

by autoregressive throughput) and acceptance length τ. Six-benchmark means weight benchmarks equally, and every result averages three evaluation seeds. Across seeds, the standard deviation of each mean τ is 0 at $T = 0$ (greedy decoding) and at most 0.02 at $T = 1$.

Objectives. DFlash and D-PACE are the baselines. All D-PAL objectives share the weight rule of Equation (6) and differ only in the local loss. We compare D-PAL[LA], D-PAL[KL], and D-PAL[Rényi] with three losses that contain an explicit TV term: D-PAL[TV], D-PAL[CE+TV], and D-PAL[KL+TV] (see Appendix A.2 for details). Table 1 uses $\rho = 0.3$ for D-PAL, the best Qwen3-4B value in Table 7, and $\rho = 0.5$ for D-PACE, the default that D-PACE reports as stable over $\rho \in [0.3, 0.7]$ (Wu et al., 2026). Ablations use $\rho = 0.5$ unless varied.

4.2 D-PAL BEATS D-PACE

Table 1 compares all objectives on Qwen3-4B. D-PAL[Rényi] improves on D-PACE at both temperatures: mean τ rises from 4.50 to 4.56 at $T = 0$ and from 4.20 to 4.28 at $T = 1$, and mean SR from 3.11 to 3.14 and from 2.79 to 2.85. The gain holds across benchmarks, since D-PAL[Rényi] has the higher τ on all 12 benchmark–temperature pairs and the higher SR on 10. The other two D-PAL objectives, D-PAL[LA] and D-PAL[KL], also beat D-PACE on both metrics at both temperatures. When we batch requests with the same checkpoints, D-PAL[Rényi] leads in τ in all ten settings and in throughput in nine (Table 2). In the separate 36K ablations at D-PACE's own $\rho = 0.5$, D-PAL[Rényi] still leads D-PACE by 0.08 in mean τ at $T = 0$ and 0.05 at $T = 1$ (Tables 3 and 7).

LA is not the best local loss. Under the same D-PAL weights, KL and Rényi reach the same or higher mean τ than LA at both temperatures (Table 1). KL leads by 0.01 at $T = 0$ and Rényi by 0.03 at $T = 1$; Rényi is our default because it also bounds overlap from above (Lemma 1).

Table 2: **D-PAL[Rényi] has the highest τ at every batch size and the highest throughput in 9 of 10 settings.** 128 MATH-500 prompts on Qwen3-4B with the 100K checkpoints; K is the inference batch size (number of requests) and TPS is output tokens per second.

	$K=1$		$K=2$		$K=4$		$K=8$		$K=16$	
Method	TPS	τ	TPS	τ	TPS	τ	TPS	τ	TPS	τ
Temperature $T=0$										
DFlash	310.24	5.46	423.17	5.48	607.93	5.46	990.60	5.49	1628.15	5.49
D-PACE	320.13	6.06	466.27	6.06	669.64	6.04	**1120.19**	6.06	1714.12	6.06
D-PAL[Rényi]	**339.73**	**6.10**	**477.46**	**6.13**	**686.75**	**6.10**	1094.14	**6.13**	**1797.82**	**6.11**
Temperature $T=1$										
DFlash	252.70	4.82	342.24	4.86	487.59	4.83	788.09	4.82	1155.27	4.83
D-PACE	274.51	5.21	362.02	5.25	515.48	5.23	786.06	5.29	1215.91	5.27
D-PAL[Rényi]	**276.68**	**5.30**	**370.04**	**5.32**	**530.42**	**5.33**	**826.42**	**5.35**	**1263.31**	**5.30**

Table 3: **Rényi beats CE under both confidence and overlap weights.** Each pair fixes the weight signal and floor and changes only the loss. Confidence with CE is D-PACE; overlap gives D-PAL[CE] and D-PAL[Rényi]. Mean τ over six benchmarks.

Loss	Signal	ρ	T=0	T=1
CE	Confidence	0.5	3.50	3.36
Rényi	Confidence	0.5	**3.58**	**3.42**
CE	Overlap	0.3	3.50	3.36
Rényi	Overlap	0.3	**3.59**	**3.43**

Table 4: **D-PAL weights beat fixed decay for all losses from baselines.** KL+TV follows the adaptive mixture of LK Losses with per-position λ_t. Static uses DFlash's decay $e^{-(t-1)/7}$; D-PAL uses $\rho=0.3$ (0.5 for CE). Mean τ over six benchmarks.

	$T=0$		$T=1$	
Local loss	Static	D-PAL	Static	D-PAL
CE	3.24	**3.50**	3.11	**3.36**
LA	3.27	**3.56**	3.15	**3.40**
CE+TV	3.40	**3.48**	3.27	**3.34**
KL+TV	3.20	**3.53**	3.08	**3.37**

Pure TV and CE+TV fall short. D-PAL[TV] collapses to $\tau=1.04$, consistent with the small TV gradient at initialization that LK Losses reports (Samarin et al., 2026). Adding CE restores training, but D-PAL[CE+TV], a DSpark- and H-Spec-style mixture, trails D-PAL[KL] by 0.18 at $T=0$ and 0.14 at $T=1$ and falls below D-PACE. Its fixed TV term keeps the nondifferentiable points at $q_{t,v}=p_{t,v}$ (Section 3.2), and the toy example in Figure 3 (Appendix A.2) shows that its CE gradient can raise TV to first order. D-PAL[KL+TV], using the adaptive mixture of losses proposed in LK Losses, shifts weight to KL at low-acceptance positions and stays within 0.04 of D-PAL[LA], D-PAL[KL], and D-PAL[Rényi].

4.3 Ablations separate the loss from the weights

Distribution fitting beats cross-entropy. Table 3 changes only the local loss, from CE to Rényi, under each weight signal. The switch raises mean τ by 0.08 at $T=0$ and 0.06 at $T=1$ under D-PACE's confidence weights ($\rho=0.5$), and by 0.09 and 0.07 under overlap weights ($\rho=0.3$). With Rényi at the same floor $\rho=0.5$, the two weight signals give similar mean τ (3.58 for both at $T=0$; 3.42 and 3.41 at $T=1$; Tables 3 and 7). We keep overlap as the weight signal because it matches rejection-sampling acceptance, reduces to D-PACE's weights for a point-mass target, and bounds argmax acceptance (Proposition 1).

D-PAL weights beat fixed decay on baseline losses. In Table 4, we compare DFlash's static weight decay against D-PAL's weight schedule across four losses from the literature: CE, used in Chen et al. (2026); LA and KL+TV, used in Samarin et al. (2026); and CE+TV, used in Cheng et al. (2026). At $T=0$, D-PAL weights raise mean τ over DFlash's schedule by 0.26 for CE, 0.29 for LA, 0.33 for KL+TV, and 0.08 for CE+TV. Gains are smaller at $T=1$ but hold for every loss.

Table 5: **D-PAL[Rényi] has the highest τ at every block size.** Mean τ over six benchmarks; three-layer drafts.

	D-PACE		D-PAL[Rényi]	
B	T=0	T=1	T=0	T=1
8	3.41	3.28	**3.48**	**3.34**
12	3.51	3.36	**3.57**	**3.42**
16	3.50	3.36	**3.58**	**3.41**

Table 6: **Deeper drafts help both methods; D-PAL[Rényi] leads at both depths.** Mean τ over six benchmarks.

	D-PACE		D-PAL[Rényi]	
Layers	T=0	T=1	T=0	T=1
3	3.50	3.36	**3.58**	**3.41**
5	3.75	3.56	**3.81**	**3.63**

Table 7: **Without the floor, D-PAL[Rényi] collapses; $\rho \in \{0.3, 0.5\}$ works best.** Mean τ over six benchmarks as the floor ρ varies.

	Qwen3-4B				Llama-3.1-8B	
	$B = 8$		$B = 16$		$B = 10$	
ρ	T=0	T=1	T=0	T=1	T=0	T=1
0	1.51	1.50	1.03	1.03	1.60	1.51
0.3	**3.51**	**3.37**	**3.59**	**3.43**	3.29	2.69
0.5	3.48	3.34	3.58	3.41	**3.31**	**2.73**
0.7	3.43	3.30	3.50	3.35	3.23	2.65
0.9	3.37	3.25	3.36	3.24	3.09	2.58

Table 8: **D-PAL[Rényi] leads on both 8B targets in SR and τ.** Means over six benchmarks; three-layer drafts.

		Llama-3.1-8B $B = 10$		Qwen3-8B $B = 16$	
Method	Metric	T=0	T=1	T=0	T=1
D-PACE	SR	2.47	1.77	2.59	2.37
	τ	3.18	2.61	3.54	3.34
D-PAL[Rényi]	SR	**2.57**	**1.84**	**2.65**	**2.40**
	τ	**3.31**	**2.73**	**3.63**	**3.39**

D-PAL[Rényi] leads at every block size and depth. Larger blocks help up to $B = 12$: the mean τ of D-PAL[Rényi] rises from $B = 8$ to $B = 12$ and levels off at $B = 16$ (Table 5). Deeper drafts help both methods, and with five layers D-PAL[Rényi] still leads D-PACE by 0.06 at $T = 0$ and 0.07 at $T = 1$ (Table 6).

The floor prevents collapse. Without the floor ($\rho = 0$), mean τ falls to 1.03–1.60, likely because an early low-overlap position drives every later prefix product toward zero (Table 7). Any floor in $\{0.3, 0.5\}$ works: the two stay within 0.04 of each other on every configuration.

4.4 Gains hold on other targets and inference settings

Gains transfer to 8B targets. D-PAL[Rényi] has the highest mean SR and τ on both 8B targets at both temperatures (Table 8). The gain is largest on Llama, where D-PAL[Rényi] leads D-PACE by 0.13 at $T = 0$ and 0.12 at $T = 1$, while D-PACE falls below Static[CE+TV].

Gains hold in thinking mode. With thinking enabled, all three D-PAL losses match or improve D-PACE in mean τ; D-PAL[Rényi] leads at $T = 0$ and D-PAL[KL] at $T = 1$ (Table 9).

5 Related Work

Parallel drafters. Medusa and DFlash propose every block position in parallel (Cai et al., 2024; Chen et al., 2026). Their per-position marginals ignore tokens drafted earlier in the block, which LiLiCorr calls the marginals problem (Rusanovsky et al., 2026). EAGLE, Hydra, Domino, and DSpark condition later proposals on earlier tokens (Li et al., 2024a; 2025; Ankner et al., 2024; Huang et al., 2026; Cheng et al., 2026). LiLiCorr reranks adjacent top-k candidates with a head co-trained with DFlash, DFlash2 adds a predecessor-conditioned selector, and DFlow carries verifier representations between rounds (Rusanovsky et al., 2026; Inco AI, 2026; Zhang et al., 2026). H-Spec replaces the drafter-side KV cache with a hybrid Mamba–attention drafter trained with CE+TV and fixed decay (Jiang et al., 2026). These methods change what the drafter conditions on; orthogonally, D-PAL changes the objective that trains the marginals and can be combined with them.

Table 9: **With thinking enabled, D-PAL[Rényi] leads mean τ at $T = 0$ and D-PAL[KL] at $T = 1$.** DFlash is the reference checkpoint (protocol in Appendix A.1).

	Acceptance length τ						
Method	GSM8K	MATH-500	HumanEval	MBPP	MT-Bench	Alpaca	Mean
Temperature $T = 0$							
DFlash	3.64	3.53	3.92	3.55	3.12	3.04	3.47
D-PACE	3.94	3.82	4.14	3.96	3.28	3.17	3.72
D-PAL[LA]	3.96	**3.91**	**4.21**	3.85	3.47	3.22	3.77
D-PAL[KL]	3.98	3.90	4.20	3.96	3.46	3.23	3.79
D-PAL[Rényi]	**4.09**	3.87	4.05	**4.02**	**3.52**	**3.31**	**3.81**
Temperature $T = 1$							
DFlash	3.21	3.10	3.08	3.01	2.69	2.62	2.95
D-PACE	3.42	3.34	3.29	3.20	2.84	2.73	3.14
D-PAL[LA]	3.44	3.35	3.29	3.18	2.86	2.73	3.14
D-PAL[KL]	**3.47**	**3.38**	**3.33**	**3.23**	**2.87**	**2.79**	**3.18**
D-PAL[Rényi]	**3.47**	3.37	3.31	3.21	**2.87**	2.77	3.17

Acceptance-aware position weighting. Several methods adapt position weights to acceptance. D-PACE weights CE by a surrogate of expected length built from draft confidence (Wu et al., 2026), following EAGLE-2's observation that draft confidence tracks acceptance (Li et al., 2024b). PARD-2 instead uses the product of the target's probabilities of the preceding recorded tokens (An et al., 2026). Other methods tie supervision to where the draft first fails: GRIFFIN propagates top-k alignment masks (Hu et al., 2025), Spec-AUF keeps CE through the first predicted failure (Yang & Li, 2026), and Whalen et al. (2026) study first-error supervision with an expected-length chain reward. VAT keeps full weight up to the first simulated rejection, restarts the decay there, and trains a verification head (Gu et al., 2026). It also finds that weighting by the exact derivative of expected length, computed from confidence, trails D-PACE's weights. D-PAL weights keep D-PACE's weighting rule and floor and use overlap, $1 - \mathrm{TV}(p, q)$, in place of confidence.

Local losses for acceptance. Other works change the local loss. DistillSpec finds that the best divergence depends on the task and decoding strategy (Zhou et al., 2024), and DART fits target-to-draft KL with fixed decay (Liu et al., 2026). Draft-OPD and AdaFlash train on draft-generated contexts with forward and reverse KL or clipped reverse KL (Lei et al., 2026; Qian et al., 2026). Closer to our setting, Bebop optimizes a normalized expected length built from TV acceptance (Li et al., 2026b). LK Losses minimizes the negative log acceptance, the same term as our LA, or an adaptive KL–TV mixture, and notes that the negative log acceptance reduces to CE for a point-mass target (Samarin et al., 2026). It keeps fixed position decay and leaves learnable or data-dependent weighting across draft heads to future work. DSpark and Uno mix CE or distillation with TV terms (Cheng et al., 2026; Sahoo et al., 2026). D-PAL puts acceptance into the weights, so the loss can be a smooth distributional fit (Section 4).

6 Conclusion

We derived D-PAL from the expected accepted length in self-speculative decoding by replacing the point-mass confidence from D-PACE (Wu et al., 2026) with the full-distribution overlap. The derivation gives acceptance-derived position weights with log acceptance as the local loss. Furthermore, we show that other losses, such as KL and Rényi-$1/2$, match or exceed the derived log acceptance in average acceptance, while enjoying theoretical guarantees and smoothness, and show they can be combined with the detached D-PAL weight schedule. Experiments across six benchmarks show up to 5% longer acceptance lengths and higher throughput than D-PACE, and up to 11% higher than DFlash (Chen et al., 2026).

Limitations and future work. D-PAL weights assume that proposals within a block do not condition on each other, as in DFlash; drafters such as DSpark or EAGLE would need weights that model this dependence (Cheng et al., 2026; Li et al., 2024a; 2025). D-PAL weights could also train the parallel logits of DFlash2 or LiLiCorr (Inco AI, 2026; Rusanovsky et al., 2026).

Reproducibility Statement

Appendix A gives training and evaluation settings. Sections 2.2–3 and Algorithm 1 specify the objectives, with supporting derivations in Appendices B.1 and B.3. Source code will accompany the submission as supplementary material.

AI Use Statement

Generative AI tools assisted with drafting and revising text. Target language models generated the training responses as described in Section 4. The authors are responsible for the final manuscript.

References

Zihao An, Taichi Liu, Ziqiong Liu, Dong Li, Ruofeng Liu, and Emad Barsoum. PARD-2: Target-aligned parallel draft model for dual-mode speculative decoding. *arXiv preprint arXiv:2605.08632*, 2026. URL https://arxiv.org/abs/2605.08632.

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. URL https://arxiv.org/abs/2402.05109.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

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. URL https://arxiv.org/abs/2401.10774.

Clément L. Canonne. A short note on an inequality between KL and TV. arXiv preprint arXiv:2202.07198, 2023. URL https://arxiv.org/abs/2202.07198.

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. URL https://arxiv.org/abs/2302.01318.

Jian Chen, Yesheng Liang, and Zhijian Liu. DFlash: Block diffusion for flash speculative decoding. *arXiv preprint arXiv:2602.06036*, 2026. URL https://arxiv.org/abs/2602.06036.

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. URL https://arxiv.org/abs/2107.03374.

Xin Cheng, Xingkai Yu, Chenze Shao, Jiashi Li, Yunfan Xiong, Yi Qian, Jiaqi Zhu, Shirong Ma, Xiaokang Zhang, Jiasheng Ye, Qinyu Chen, Chengqi Deng, Jiping Yu, Damai Dai, Zhengyan Zhang, Yixuan Wei, Yixuan Tan, Wenkai Yang, Runxin Xu, Yu Wu, Zhean Xu, Xuanyu Wang, Muyang Chen, Rui Tian, Xiao Bi, Zhewen Hao, Shaoyuan Chen, Huanqi Cao, Wentao Zhang, Anyi Xu, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. DSpark: Confidence-scheduled speculative decoding with semi-autoregressive generation. *arXiv preprint arXiv:2607.05147*, 2026. URL https://arxiv.org/abs/2607.05147.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018. URL https://arxiv.org/abs/1812.02224.

Christopher A. Fuchs and Jeroen van de Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, 45(4):1216–1227, 1999. doi: 10.1109/18.761271. URL https://arxiv.org/abs/quant-ph/9712042.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Geonmo Gu, Byeongho Heo, HeeJae Jun, Yoohoon Kang, Sangmin Lee, Sangdoo Yun, and Dongyoon Han. Verification-aware training for speculative decoding. *arXiv preprint arXiv:2608.30135*, 2026. URL https://arxiv.org/abs/2608.30135.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2103.03874.

Shijing Hu, Jingyang Li, Xingyu Xie, Zhihui Lu, Kim-Chuan Toh, and Pan Zhou. GRIFFIN: Effective token alignment for faster speculative decoding. *arXiv preprint arXiv:2502.11018*, 2025. URL https://arxiv.org/abs/2502.11018.

Jianuo Huang, Yaojie Zhang, Qituan Zhang, Hao Lin, Hanlin Xu, and Linfeng Zhang. Domino: Decoupling causal modeling from autoregressive drafting in speculative decoding. *arXiv preprint arXiv:2605.29707*, 2026. URL https://arxiv.org/abs/2605.29707.

Inco AI. DFlash 2: Keep drafting parallel, August 2026. URL https://inco.ai/blog/dflash2/.

Weifan Jiang, Krishna Teja Chitty-Venkata, Megan Flynn, Reed Meyerson, Zhenting Qi, Tianyu Wu, Eldar Kurtic, Minlan Yu, and Alexandre Marques. H-Spec: Parallel speculative decoding without a drafter-side KV cache. *arXiv preprint arXiv:2609.24197*, 2026. URL https://arxiv.org/abs/2609.24197.

Haodi Lei, Yafu Li, Haoran Zhang, Shunkai Zhang, Qianjia Cheng, Xiaoye Qu, Ganqu Cui, Bowen Zhou, Ning Ding, Yun Luo, and Yu Cheng. Draft-OPD: On-policy distillation for speculative draft models. *arXiv preprint arXiv:2605.29343*, 2026. URL https://arxiv.org/abs/2605.29343.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2211.17192.

Shenggui Li, Chao Wang, Yikai Zhu, Yubo Wang, Fan Yin, Shuai Shi, Yefei Chen, Xiaomin Dong, Qiaoling Chen, Jin Pan, Ji Li, Laixin Xie, Yineng Zhang, Lei Yu, Yonggang Wen, Ivor Tsang, and Tianwei Zhang. SpecForge: A flexible and efficient open-source training framework for speculative decoding. *arXiv preprint arXiv:2603.18567*, 2026a. URL https://arxiv.org/abs/2603.18567.

Yucheng Li, Huiqiang Jiang, Yang Xu, Jianxin Yang, Yi Zhang, et al. Breaking entropy bounds: Accelerating RL training via MTP with rejection sampling. *arXiv preprint arXiv:2606.12370*, 2026b. URL https://arxiv.org/abs/2606.12370.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024a. URL https://arxiv.org/abs/2401.15077.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-2: Faster inference of language models with dynamic draft trees. *arXiv preprint arXiv:2406.16858*, 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*, 2025. URL https://arxiv.org/abs/2503.01840.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050.

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. URL https://arxiv.org/abs/2601.19278.

Yu-Yang Qian, Hao-Cong Wu, Chen Chen, Jiacheng Sun, Zhenhua Dong, Peng Zhao, and Zhi-Hua Zhou. AdaFlash: Adaptive speculative decoding via on-policy distilled diffusion drafters. *arXiv preprint arXiv:2607.19223*, 2026. URL https://arxiv.org/abs/2607.19223.

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, 1961.

Matan Rusanovsky, Yoav Miron, Roy Uziel, Omer Belhasin, Ran Zilberstein, Maor Ashkenazi, and Michael Elad. LiLiCorr: Lightweight likelihood correlation of parallel drafts for speculative decoding. *arXiv preprint arXiv:2608.20530*, 2026. URL https://arxiv.org/abs/2608.20530.

Subham Sekhar Sahoo, Lingjie Chen, Khiem Pham, Jonathan Geuter, Chaitanya Dwivedi, Varad Pimpalkhute, Yash Akhauri, Alexander Moreno, Mikhail Yurochkin, Zhenting Wang, Mostafa Elhoushi, Nolan Dey, Shane Bergsma, Joel Hestness, John Thickstun, Eric Xing, and Zhengzhong Liu. Unlocking lossless speedups in LLMs via discrete diffusion. arXiv preprint arXiv:2609.04010, 2026. URL https://arxiv.org/abs/2609.04010.

Alexander Samarin, Sergei Krutikov, Anton Shevtsov, Sergei Skvortsov, Filipp Fisin, and Alexander Golubev. LK Losses: Direct acceptance rate optimization for speculative decoding. *arXiv preprint arXiv:2602.23881*, 2026. URL https://arxiv.org/abs/2602.23881.

SGLang Contributors. SGLang v0.5.19 DFlash speculative verification. https://github.com/sgl-project/sglang/blob/v0.5.19/python/sglang/srt/speculative/dflash_utils.py, 2026. Source code, lines 775–995.

SpecForge Contributors. SpecForge unified training recipe catalog. Software documentation, 2026. URL https://github.com/sgl-project/SpecForge/blob/main/examples/configs/README.md. Accessed September 23, 2026.

Speculators Contributors. Speculators training command reference. Software documentation, 2026. URL https://docs.vllm.ai/projects/speculators/en/stable/cli/train/. Accessed September 23, 2026.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpaca: A strong, replicable instruction-following model. Stanford Center for Research on Foundation Models, 2023. URL https://crfm.stanford.edu/2023/03/13/alpaca.html.

TorchSpec Contributors. TorchSpec DFlash training guide. Software documentation, 2026. URL https://github.com/lightseekorg/TorchSpec/blob/main/docs/dflash/README.md. Accessed September 23, 2026.

Tim van Erven and Peter Harremoës. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014. doi: 10.1109/TIT.2014.2320500. URL https://arxiv.org/abs/1206.2459.

Lexington Whalen, Yuki Ito, and Ryo Sakamoto. Teaching diffusion to speculate left-to-right. *arXiv preprint arXiv:2606.11552*, 2026. URL https://arxiv.org/abs/2606.11552.

Tianyu Wu, Yu Yao, Zhenting Qi, Han Zheng, Zhuohan Wang, Haoran Ma, Lawrence Liao, Himabindu Lakkaraju, Ju Li, and Yilun Du. D-PACE: Dynamic position-aware cross-entropy for parallel speculative drafting. *arXiv preprint arXiv:2605.18810*, 2026. URL https://arxiv.org/abs/2605.18810.

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. URL https://arxiv.org/abs/2505.09388.

Tianjian Yang and Meng Li. Spec-AUF: Accept-until-fail training under train-inference misalignment for masked block drafters. *arXiv preprint arXiv:2607.01893*, 2026. URL https://arxiv.org/abs/2607.01893.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf.

Yaojie Zhang, Linfeng Zhang, Bin Cui, and Xupeng Miao. DFlow: Enabling verifier information flow in block diffusion speculative decoding. *arXiv preprint arXiv:2609.06498*, 2026. URL https://arxiv.org/abs/2609.06498.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2306.05685.

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. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.08461.

A Experimental Settings and Loss Definitions

A.1 Training and evaluation

Table 1 uses the Qwen3-4B-Instruct-100K dataset released by the DFlash authors, which D-PACE also uses for its main Qwen3-4B experiments (Chen et al., 2026; Wu et al., 2026). Of its 99,987 rows, 99,171 are valid training sequences. We randomly sample ShareGPT prompts with SpecForge for the roughly 36K-example runs. All training responses in this paper are generated by the frozen target at temperature 0.7; Qwen3 uses non-thinking mode. Methods share data within each comparison (settings in Table 10). Results in Tables 1–2 average evaluation seeds 980406, 980407, and 980408. Across evaluation seeds, the standard deviation of each six-benchmark mean τ is 0 at $T = 0$ and at most 0.02 at $T = 1$. Every configuration uses training seed 42.

Table 10: **Training and evaluation settings.** Each block includes one anchor token excluded from the loss. Both draft depths use the same three target feature layers.

Training	
Optimizer / epochs	AdamW / 6
Learning rate / schedule	6×10^{-4}; cosine decay, 4% warm-up
Global batch / seed	8 (Tables 1 and 2), 4 (other runs); seed 42
Sequence / anchor limits	3072 tokens / 512 anchors
Default block size	Qwen: $B = 16$; Llama: $B = 10$
Loss temperature	1 (target and draft)
Evaluation	
Backend	SGLang 0.5.19 DFLASH
Evaluation temperatures	$T = 0$, $T = 1$
Generation limit	2048 new tokens

Compute. We estimate that running all training experiments on NVIDIA H200 GPUs would require approximately 840 GPU-hours, including 138 GPU-hours for the eight 100K runs. We run all evaluations on NVIDIA L40S GPUs.

Target and draft distributions. The frozen target output head maps cached final hidden states to full-vocabulary logits; softmax gives p_t. At each recorded position, the draft predicts q_t from the anchor context and a masked block. It does not see the full prefix available to the target at later positions. Both loss softmaxes use temperature one; we generate labels at temperature 0.7.

Inference protocol. SGLang 0.5.19 DFLASH uses argmax draft proposals at both temperatures. At $T = 0$, verification retains the target-argmax prefix; at $T = 1$, it applies target-only stochastic acceptance and rejection correction, which accepts the deterministic proposal $\hat{y}_t$ with probability $p_t(\hat{y}_t)$ (SGLang Contributors, 2026). We disable top-k and top-p truncation and set SGLang's single-token and accumulated acceptance thresholds to one. We measure τ through verification; overlap serves as a training proxy.

The grouped-request experiment in Table 2 uses 128 fixed MATH-500 prompts and one L40S. All inference batch sizes, including $K = 1$, use grouped-request inference, so its throughput is not directly comparable with the single-request results in Table 1. It evaluates the 100K checkpoints of DFlash, D-PACE, and D-PAL[Rényi] from Table 1.

Evaluation subsets. We use 128 examples each from GSM8K, MATH-500, and MBPP, 164 from HumanEval, and 80 first-turn MT-Bench prompts, with subset seed 42. Alpaca uses a fixed list of 128 single-turn prompts, supplied as `alpaca_128.jsonl`. We use the same prompts across methods and temperatures within each target.

Thinking-mode comparison. Table 9 uses a separate set of three-layer, $B = 16$ checkpoints on the same six benchmark subsets, with thinking enabled. The D-PACE and D-PAL checkpoints use $\rho = 0.3$. The context limit is 16384 tokens and the generation limit is 8192; both reasoning and final-answer tokens count toward the metrics. Each run uses one L40S and one active request. We retain responses that reach the generation limit.

A.2 Local losses and position weights

Table 11 lists the local fitting terms, with recorded target token $y_t^\star$ and training overlap $\widehat{\alpha}_t = 1 - \mathrm{TV}(p_t, q_t)$. All D-PAL objectives use the detached weights in Equation (6) and differ only in their local loss. Divergences and overlap use the full vocabulary.

Table 11: **Local losses compared in this paper.** D-PAL[KL+TV] computes $\lambda_t = \exp(-3\widehat{\alpha}_t)$ for each block and position and detaches it during optimization.

Objective	Local loss ℓ_t
Cross-entropy	$\ell_{\mathrm{CE},t} = -\log q_t(y_t^\star)$
D-PAL[LA]	$-\log[1 - \mathrm{TV}(p_t, q_t)]$
D-PAL[KL]	$\mathrm{KL}(p_t \Vert q_t)$
D-PAL[Rényi]	$D_{1/2}(p_t \Vert q_t)$
D-PAL[TV]	$\mathrm{TV}(p_t, q_t)$
D-PAL[CE+TV]	$0.1\ell_{\mathrm{CE},t} + 0.9\,\mathrm{TV}(p_t, q_t)$
D-PAL[KL+TV]	$\lambda_t\,\mathrm{KL}(p_t \Vert q_t) + (1 - \lambda_t)\,\mathrm{TV}(p_t, q_t)$

DSpark's $0.1\mathcal{L}_{\mathrm{CE}} + 0.9\Vert p - q\Vert_1$ has twice the TV coefficient of our CE+TV loss. D-PAL[KL+TV] computes λ_t at each position, whereas LK Losses aggregates acceptance across sequence and batch before setting its hybrid coefficient (Samarin et al., 2026).

Static weights use $\exp(-(t-1)/\gamma)$, with $\gamma = 4, 5, 6, 7$ for $B = 8, 10, 12, 16$. D-PACE smooths confidence with $\rho = 0.5$. D-PAL smooths $\widehat{\alpha}_t$ with $\rho = 0.3$ in Table 1 and with $\rho = 0.5$ in the ablations unless varied; D-PAL[CE] uses $\rho = 0.3$. We average blocks within sequences, then sequences, without normalizing by the weight sum.

Implementation. The two weight scans cost $O(D_b)$; vocabulary sums cost $O(D_b V)$. We compute the Rényi loss in 32-bit floating point (FP32) using log-softmax and logsumexp.

Gradient conflict in TV mixtures. A fixed positive TV coefficient leaves a mixture nondifferentiable, and the auxiliary gradient can oppose the TV gradient when the terms share parameters (Du et al., 2018; Yu et al., 2020). Let $g_T = \nabla_\theta \mathrm{TV}$ and $g_F = \nabla_\theta \mathcal{L}_F$. At differentiable points, a stochastic gradient descent step of size η on $a\,\mathrm{TV} + b\mathcal{L}_F$ changes TV by

$$\delta\,\mathrm{TV} = -\eta\big(a\Vert g_T\Vert_2^2 + b\,g_T^\top g_F\big) + o(\eta), \tag{11}$$

so the mixed step raises TV to first order when $a\Vert g_T\Vert_2^2 + b\,g_T^\top g_F < 0$.

For target $p = (0.45, 0.35, 0.20)$ and draft $q = (0.55, 0.01, 0.44)$, vary two draft logits and fix the third at zero. With hard-target CE for token 1 and $(a, b) = (0.9, 0.1)$, $g_T^\top(0.1 g_{\mathrm{CE}} + 0.9 g_T) < 0$: descent on the mixed loss raises TV to first order (Figure 3).

KL and TV gradients can also oppose, which can affect D-PAL[KL+TV]. For $p = (0.4, 0.5, 0.1)$ and $q(\theta) = \mathrm{softmax}(\log(0.1, 0.6, 0.3) + \theta(0, 1, -1))$, $d\,\mathrm{TV}/d\theta = 0.03$ and $d\,\mathrm{KL}/d\theta = -0.10$ at $\theta = 0$: the KL and TV gradients oppose. With detached position weights, Equation (11) applies to the weighted sum of these gradients over positions.

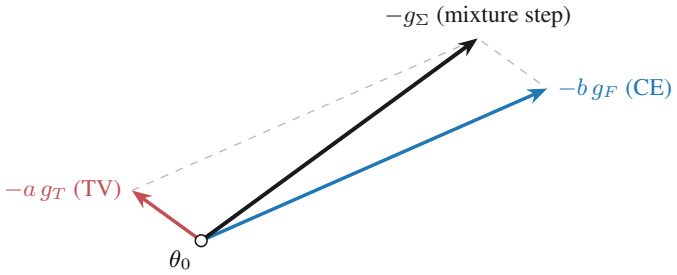


Figure 3: **A CE+TV descent step can raise TV.** Negative weighted gradients of TV ($a = 0.9$) and CE ($b = 0.1$) at one parameter point of the three-token example, on one shared scale and rotated 25° for display. The two gradients have a negative inner product, and the mixture step $-g_\Sigma$ raises TV to first order ($g_T^\top g_\Sigma < 0$).

B Additional Derivations

B.1 Rényi fitting gradient

Fix p and let $q = \mathrm{softmax}(z)$. Using $\partial q_j / \partial z_v = q_j(\mathbf{1}_{j=v} - q_v)$,

$$\frac{\partial \mathrm{BC}(p,q)}{\partial z_v} = \frac{1}{2}\big(\sqrt{p_v q_v} - q_v \mathrm{BC}(p,q)\big), \qquad \frac{\partial D_{1/2}}{\partial z_v} = -\frac{2}{\mathrm{BC}(p,q)}\frac{\partial \mathrm{BC}(p,q)}{\partial z_v} = q_v - r_v.$$

Here $r_v = \sqrt{p_v q_v}/\mathrm{BC}(p,q)$ is the normalized geometric mean.

Comparison with KL. Rényi divergence is nondecreasing in order, so $D_{1/2}(p\|q) \le \mathrm{KL}(p\|q)$ (van Erven & Harremoës, 2014). Bretagnolle–Huber gives $\alpha(p,q) \ge 1 - \sqrt{1 - e^{-\mathrm{KL}(p\|q)}}$ (Canonne, 2023). For the same p, q, Equation (10) gives an overlap lower bound at least as high as the KL bound.

B.2 Proofs of Proposition 1 and Lemma 1

Proof of Lemma 1. The upper bound follows from $\min\{p_j, q_j\} \le \sqrt{p_j q_j}$. For the lower bound, set $u_j = \sqrt{p_j}$ and $v_j = \sqrt{q_j}$. Both have unit norm, and $u^\top v = \mathrm{BC}(p,q)$:

$$\begin{aligned} 2\,\mathrm{TV}(p,q) &= \sum_j |(u_j - v_j)(u_j + v_j)| \\ &\le \|u - v\|_2 \, \|u + v\|_2 = 2\sqrt{1 - \mathrm{BC}(p,q)^2} = 2\sqrt{1 - e^{-D_{1/2}(p\|q)}}. \end{aligned}$$

Hence $\alpha(p,q) = 1 - \mathrm{TV}(p,q) \ge 1 - \sqrt{1 - e^{-D_{1/2}(p\|q)}}$.

Proof of Proposition 1. Write $p = p_t$, $q = q_t$, $\Delta = \Delta_t$, $\hat{y} = \arg\max_v q_v$, and $v^\star = \arg\max_v p_v$. (i) Suppose $\hat{y} = j \ne v^\star$, so $q_j \ge q_{v^\star}$. Then

$$2\,\mathrm{TV}(p,q) \ge (p_{v^\star} - q_{v^\star}) + (q_j - p_j) = (p_{v^\star} - p_j) + (q_j - q_{v^\star}) \ge \Delta,$$

which contradicts $\mathrm{TV}(p,q) < \Delta/2$. (ii) Since $\mathrm{TV}(p,q) = \sum_v (q_v - p_v)_+ \ge q_{\hat{y}} - p_{\hat{y}}$, we have $p_{\hat{y}} \ge \max_v q_v - \mathrm{TV}(p,q)$.

B.3 TV gradients and position weights

TV coefficients. Fix the target model and target-generated training contexts. Differentiating the prefix sum of training overlaps $\widehat{\alpha}_t = 1 - \mathrm{TV}(p_t, q_t)$ gives

$$\nabla_\theta \sum_{k=1}^{D} \prod_{i=1}^{k} \widehat{\alpha}_i = \sum_{t=1}^{D} \left[\sum_{k=t}^{D} \prod_{\substack{1 \le i \le k \\ i \ne t}} \widehat{\alpha}_i \right] \nabla_\theta \widehat{\alpha}_t. \tag{12}$$

Multiplying this coefficient by $\widehat{\alpha}_t$ gives the relative-contribution form in Equation (3), evaluated at training overlaps.

Gradient of the smoothed prefix sum. For positive $s_t = \rho + (1-\rho)\widehat{\alpha}_t$, direct differentiation of the smoothed prefix sum gives

$$\nabla_\theta \sum_{k=1}^{D} \prod_{i=1}^{k} s_i = \sum_{t=1}^{D} \frac{1-\rho}{s_t} w_t \, \nabla_\theta \widehat{\alpha}_t.$$

At $\rho = 0$, $s_t = \widehat{\alpha}_t$ and gradient descent on D-PAL[LA] follows this direction; for $\rho > 0$, the factor $(1-\rho)/s_t$ replaces $1/\widehat{\alpha}_t$, and we treat the floor as a regularizer, as D-PACE does.