D-VOLT: TRAINING SPECULATIVE DRAFTERS WITH TOKEN-CONDITIONAL ACCEPTANCE VALUES

Anonymous authors
Paper under double-blind review

ABSTRACT

Speculative decoding accelerates language model inference by verifying several draft tokens in a single target-model forward pass. Training a lightweight drafter requires allocating its limited capacity to improve the expected number of accepted draft tokens. Existing approaches improve local distribution matching or reweight losses across positions, but candidate tokens at the same prefix can enable different expected numbers of subsequent acceptances. We identify a limitation of weighting a local acceptance objective by a detached continuation weight shared across candidate tokens at the same prefix and position. Even when the weight uses the exact mean continuation value conditional on acceptance, the resulting objective can favor updates that increase one-step acceptance probability while decreasing expected accepted length. We propose D-VOLT (Drafting via Value-Optimized Lookahead Training), which assigns training credit using token-conditional continuation values. We derive a Bellman recursion over complete prefix histories and an exact gradient decomposition of expected accepted length, in which each candidate's credit combines prefix reachability with its immediate acceptance reward and continuation value. For practical training, we approximate continuation values by dynamic programming over a compact top-K candidate graph that shares sampled earlier histories across branches. We combine full-vocabulary total-variation matching with a candidate-specific continuation reward, retaining the drafter architecture and standard verification procedure. Experiments with DSpark drafters on BBH, MMLU-Pro, WinoGrande, and GSM8K at decoding temperatures $T = 0$ and $T = 1$ show that D-VOLT improves average accepted length by 4.05%–9.74% over the DSpark baseline and by 2.04%–6.46% over D-PACE.

1 INTRODUCTION

Speculative decoding accelerates autoregressive language model inference by using a lightweight drafter to propose several tokens that a larger target model verifies in parallel. With rejection sampling, this procedure preserves the target model's output distribution while amortizing the cost of a target forward pass over multiple output tokens (Leviathan et al., 2023; Chen et al., 2023). Recent methods also target efficient block drafting: DFlash generates a block in a single forward pass using a target-conditioned diffusion model (Chen et al., 2026); DSpark combines a parallel backbone with a lightweight sequential head to model dependencies within the block (Cheng et al., 2026). Acceptance proceeds only up to the first rejection, making expected accepted length central to efficiency at fixed per-block drafting and verification costs.

To keep drafting inexpensive, the drafter must remain compact. Its limited capacity makes it difficult to match the target distribution across all prefixes. The remaining mismatches have unequal effects on accepted length: correcting one may recover only a single accepted token, whereas correcting another may enable a longer accepted suffix. Thus, reducing average distributional mismatch need not yield the greatest improvement in expected accepted length. Training objectives should therefore account for the acceptance gains enabled by each distributional improvement.

Two complementary directions seek to better align drafter training with speculative decoding. One adjusts *how training losses are weighted across positions*. D-PACE derives adaptive position weights from a draft-confidence surrogate of expected accepted length (Wu et al., 2026). The other

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.

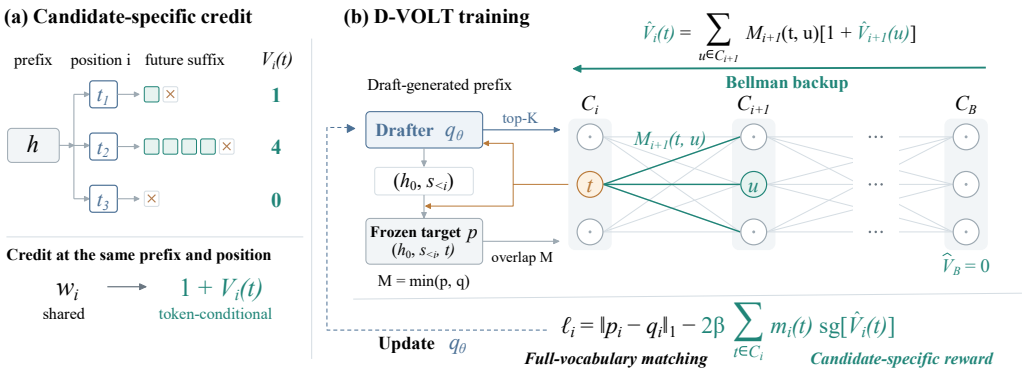

Figure 1: **Candidate-specific credit and D-VOLT training.** (a) Candidates at the same prefix and position can have different continuation values (illustrative values: 1, 4, and 0), motivating candidate-specific credit rather than a weight shared across candidates. (b) Top-K candidate sets fixed across branches and shared sampled earlier histories define a compact graph on draft-generated prefixes. Backward dynamic programming estimates continuation values from target/draft probability overlaps. Detached value estimates weight a candidate-specific continuation reward alongside full-vocabulary total-variation matching. The target remains frozen.

reduces the *mismatch between training inputs and those encountered during drafting*. Draft-OPD uses target feedback on both accepted and rejected draft proposals to train on draft-generated prefixes (Lei et al., 2026). EAGLE-3's training-time test feeds the drafter's own intermediate outputs back into training to better match multi-step inference (Li et al., 2025). A complementary question is how training credit should differ among candidate tokens at the same prefix and position.

Our starting observation is that *candidate tokens at the same prefix and position can have different continuation values*. A token's continuation value is the expected number of additional draft tokens accepted within the current block after that token is accepted. In a diagnostic using 100 GSM8K prompts and a seven-position DSpark drafter, we obtain Monte Carlo estimates of continuation values for the top-16 candidates at each fixed sampled prefix. Across nonterminal positions, the mean within-prefix standard deviation ranges from 24.5% to 34.8% of the remaining horizon (Section 3.1). We also construct an oracle that reallocates accepted probability mass across recorded prefixes and candidate tokens while preserving mean acceptance at each position. With recorded prefixes and continuation-value estimates held fixed, feasible draft distributions improve estimated local return over the recorded top-16 support by 5.07% (Section 3.2).

This variation exposes a limitation of a detached continuation weight shared across candidates at the same prefix and position. Even when the weight uses the exact mean continuation value conditional on acceptance, a local objective can favor higher one-step acceptance probability while decreasing expected accepted length. Section 3.3 constructs such an update with fixed target distributions and continuation values; it also lowers target-weighted cross-entropy. The limitation concerns the granularity of continuation credit.

We propose D-VOLT (*Drafting via Value-Optimized Lookahead Training*), which assigns training credit using token-conditional continuation values. We derive an exact gradient decomposition of expected accepted length over the full vocabulary and complete prefix tree. For practical training, we evaluate the next-token distributions of the target and drafter after appending selected candidates at nonterminal positions, then estimate continuation values by backward dynamic programming over a compact top-K graph with shared sampled earlier histories. The objective combines full-vocabulary total-variation matching with a candidate-specific continuation reward, preserving the deployed drafter architecture and standard verification procedure.

Our main contributions are:

- **Candidate-value variation and allocation.** We measure variation in estimated continuation values among candidates at fixed prefixes and quantify the benefit of reallocating acceptance while

preserving its mean at each position. With prefixes and continuation-value estimates fixed, feasible draft distributions improve estimated local return over the recorded top-16 support by 5.07%.

- **Candidate-specific credit analysis.** We identify a limitation of detached weights shared across candidate tokens, even when these weights use exact mean continuation values. We derive an exact gradient decomposition of expected accepted length that combines prefix reachability with each candidate's immediate acceptance reward and continuation value.
- **A practical value-based training method.** We introduce D-VOLT, combining full-vocabulary total-variation matching with a candidate-specific continuation reward on draft-generated prefixes. In DSpark experiments across four benchmarks at $T = 0$ and $T = 1$, D-VOLT improves average accepted length by 4.05%–9.74% over the DSpark baseline and by 2.04%–6.46% over D-PACE.

2 PRELIMINARIES

Speculative decoding and accepted length. Given a verified prefix h_0, whose final token serves as the anchor, a drafter q_θ proposes a block of B tokens from vocabulary $\mathcal{V}$, which a frozen target p verifies in one forward pass. At history h, comprising h_0 and the preceding draft tokens, a proposed token t is accepted with probability $\min\{1, p(t \mid h)/q_\theta(t \mid h)\}$, provided all preceding draft tokens in the block have been accepted. At the first rejection, the rejected token and all subsequent draft tokens are discarded (Leviathan et al., 2023; Chen et al., 2023). The accepted length $X \in \{0, \dots, B\}$ counts the consecutive accepted draft tokens from the start of the block, excluding the correction or bonus token. The expectation $\mathbb{E}[X \mid h_0]$ depends on both the acceptance probability at each history and which histories are reached through earlier acceptances.

Local losses and position weights. At position i, let p_i and q_i denote the target and draft distributions, y_i the supervision token, and $c_i = q_i(y_i)$ its draft probability. Conditioning inputs and dependence on θ are omitted where clear. Local objectives take the form $\mathcal{L} = \sum_{i=1}^{B} w_i \ell_i$, with position weights w_i and losses such as cross-entropy $-\log c_i$, forward KL $D_{\mathrm{KL}}(p_i \| q_i)$, or total variation $\frac{1}{2}\|p_i - q_i\|_1$. DFlash and DSpark use fixed exponentially decaying position weights (Chen et al., 2026; Cheng et al., 2026). LK Losses uses negative log acceptance probability or an adaptive KL/TV mixture (Samarin et al., 2026). PARD-2 weights local losses using cumulative target confidence as a proxy for prefix reachability (An et al., 2026). D-PACE derives example-dependent weights from a draft-confidence surrogate of expected accepted length (Wu et al., 2026). Writing sg for stop-gradient, its unsmoothed formulation is

$$\widetilde{S} = \sum_{k=1}^{B} \prod_{i=1}^{k} c_i, \qquad w_j = \mathrm{sg}\left[\sum_{k=j}^{B} \prod_{i=1}^{k} c_i\right], \qquad \mathcal{L}_{\text{D-PACE}} = \sum_{j=1}^{B} w_j(-\log c_j). \tag{1}$$

For fixed inputs and supervision tokens, $\nabla_\theta \mathcal{L}_{\text{D-PACE}} = -\nabla_\theta \widetilde{S}$; the weights combine prefix confidence with a confidence-based continuation factor.

3 ANALYSIS: ACCEPTANCE VALUE BEYOND POSITION

At prefix h and position i, define the continuation value $V_i(h, t)$ as the expected number of additional draft tokens accepted within the current block after accepting candidate t, under the current draft policy. Conditional on reaching h through earlier acceptances, the expected number of draft tokens accepted from position i onward is

$$J_i(h) = \sum_{t \in \mathcal{V}} m_i(h, t)\big(1 + V_i(h, t)\big), \qquad m_i(h, t) = \min\{p_i(t \mid h), q_i(t \mid h)\}. \tag{2}$$

3.1 WITHIN-PREFIX VARIATION IN CANDIDATE VALUE ESTIMATES

Using 100 GSM8K prompts, we collect 3,380 seven-position draft blocks with a five-layer DSpark drafter equipped with a Markov head and Qwen3-8B as the target (Cheng et al., 2026; Yang et al.,

2025; Cobbe et al., 2021). Diagnostic plots index positions within each block from 0 to 6, corresponding to $i = 1, \ldots, B$ in the equations.

At each fixed sampled prefix, we estimate continuation values for the 16 candidates with the highest draft probabilities. For each candidate, we draw 16 Monte Carlo (MC) suffix trajectories from the draft distribution after appending that candidate, retaining the full history of each branch (Figure 2a,b). We denote the resulting continuation-value estimates by $\widehat{V}_i(h, t)$.

Across the 20,280 nonterminal prefix–position records (positions 0–5), the mean within-record standard deviation of estimated continuation values across the 16 candidates is 0.937 tokens. After normalizing each standard deviation by the remaining horizon and averaging within each position, the resulting means range from 24.5% to 34.8%. Appendix A provides the estimation protocol and uncertainty analysis.

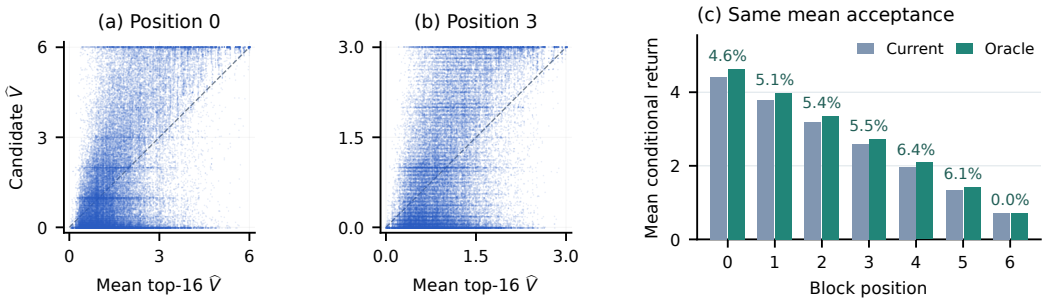

Figure 2: **Continuation-value variation and allocation headroom.** (a,b) Each point pairs a candidate's estimated continuation value (vertical axis) with the mean across the 16 candidates at the same prefix (horizontal axis). Vertical deviations from $y = x$ show within-prefix variation (54,080 points per panel). (c) Mean return over the recorded top-16 candidates before and after pooled oracle reallocation, preserving mean acceptance at each position. Returns include the current token; labels show relative gains.

3.2 Allocation Headroom Within and Across Prefixes

At each position, we ask whether reallocating accepted probability mass can improve estimated local return while preserving mean acceptance. We compare two oracle relaxations: one reallocates mass only among candidates at each prefix, while the other also allows reallocation across prefixes. Both hold the recorded prefixes, target distributions, and estimated continuation values fixed.

Within-prefix allocation. For each recorded prefix h at position i, let $C_i(h)$ contain its top-16 draft candidates. We reallocate accepted probability mass within $C_i(h)$ while preserving its total:

$$\max_{0 \le z_{h,t} \le p_i(t|h)} \sum_{t \in C_i(h)} z_{h,t}\big(1 + \widehat{V}_i(h, t)\big) \quad \text{s.t.} \quad \sum_{t \in C_i(h)} z_{h,t} = \sum_{t \in C_i(h)} m_i(h, t),$$

where $z_{h,t}$ is the accepted probability mass allocated to candidate t.

Since the total immediate reward is fixed, a feasible transfer of mass δ from candidate a to b changes return by $\delta(\widehat{V}_i(h, b) - \widehat{V}_i(h, a))$. The optimum therefore prioritizes candidates with higher estimated continuation values, subject to their target-probability caps. In our diagnostic, this relaxation improves mean return over the recorded candidates at every nonterminal position (Appendix B).

Across-prefix allocation. We next pool the recorded prefixes at each position, replacing the per-prefix constraint with

$$\sum_{h} \sum_{t \in C_i(h)} z_{h,t} = \sum_{h} \sum_{t \in C_i(h)} m_i(h, t).$$

This allows reallocation both among candidates and across prefixes while preserving mean acceptance at each position. In our diagnostic, the pooled relaxation improves mean return over the recorded candidates by 4.57%–6.36% at nonterminal positions. Averaged across all records, including terminal positions, this return increases from 2.5731 to 2.7042 (+5.10%). We also construct normalized draft distributions satisfying the same pooled constraint and leaving the vocabulary tail unchanged, retaining a 5.07% gain (Appendix B).

These fixed-prefix diagnostics quantify local allocation headroom, not changes in block-level expected accepted length.

3.3 What a Shared Continuation Weight Omits

At a fixed prefix and position, consider a local change in the draft distribution while holding the target distribution and continuation values fixed. Omitting i, h for brevity, let $A = \sum_t m_t > 0$ be the one-step acceptance probability and $\widetilde{V} = \sum_t m_t V_t / A$ the mean continuation value conditional on acceptance. Then

$$J = A(1 + \widetilde{V}), \qquad \mathrm{d}J = (1 + \widetilde{V})\,\mathrm{d}A + A\,\mathrm{d}\widetilde{V}. \tag{3}$$

The local acceptance objective uses $1 + \widetilde{V}$ as a shared weight across candidate tokens. With this weight detached, the objective's differential contains only $(1+\widetilde{V})\,\mathrm{d}A$, omitting $A\,\mathrm{d}\widetilde{V}$, the contribution from changes in the candidate distribution conditional on acceptance. This term is omitted even when $\widetilde{V}$ is computed exactly.

Figure 3 illustrates a feasible change to the draft distribution with the target distribution and continuation values held fixed. The update decreases accepted probability mass on a candidate with a higher continuation value while increasing it on a lower-value candidate. One-step acceptance probability rises from 0.80 to 0.85, and both total variation distance and target-weighted cross-entropy decrease, yet expected accepted length falls from 2.70 to 2.65. A shared positive detached weight preserves this preference for both local losses.

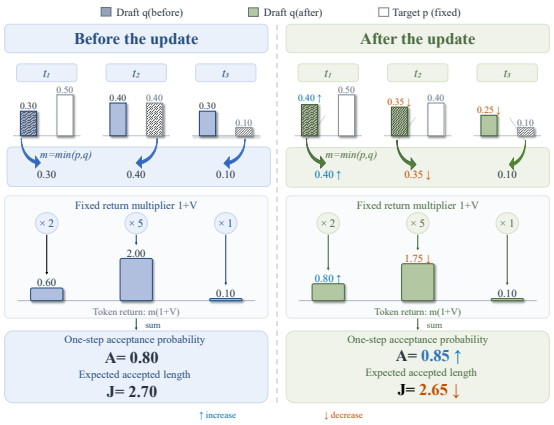

Figure 3: **Higher acceptance, lower return.** The target distribution p and continuation values V are held fixed. Accepted probability mass increases by 0.10 on t_1, decreases by 0.05 on t_2, and remains unchanged on t_3. The return multipliers $1 + V$ are 2 for t_1 and 5 for t_2, so the 0.20 gain from t_1 is outweighed by the 0.25 loss from t_2: $\Delta A = 0.05$, yet $\Delta J = -0.05$. Returns include the current draft token.

With continuation values held fixed, the omitted term vanishes when all candidates have the same continuation value. When values differ, candidate-specific credit can capture effects of reallocating accepted probability mass that a shared weight misses (Figure 1(a)). D-PACE's unsmoothed gradient identity for its own confidence surrogate remains valid (Wu et al., 2026). Appendix C gives the feasible construction and full calculation.

4 D-VOLT

Figure 1 provides an overview of candidate-specific credit and D-VOLT training. We first derive an exact gradient decomposition of expected accepted length, then introduce a practical training objective using continuation values estimated over a compact candidate graph.

4.1 Exact Acceptance Values and Credit Assignment

Fix a verified prefix h_0, treated as independent of θ, and a block of B draft positions. We work with the full vocabulary $\mathcal{V}$ and the complete prefix tree within the block. Accepting candidate t contributes one accepted draft token and extends the history to (h, t); rejection stops further draft-token acceptance within the block. Using $J_i(h)$ and $m_i(h, t)$ from Eq. 2, we obtain the Bellman recursion

$$J_i(h) = \sum_{t \in \mathcal{V}} m_i(h, t)\big[1 + V_i(h, t)\big], \qquad V_i(h, t) = J_{i+1}(h, t), \qquad J_{B+1} = 0. \tag{4}$$

Thus $J_1(h_0) = \mathbb{E}[X \mid h_0]$ exactly. Here, continuation values depend on θ through subsequent draft distributions. The recursion retains complete branch histories and requires no independence assumption between positions.

Let $\mathcal{H}_{i-1}$ contain all prefixes formed by appending $i-1$ draft tokens to h_0. Define prefix reachability $\rho_i(h)$ as the probability of proposing and accepting all tokens in this extension. With $\rho_1(h_0) = 1$, recursively differentiating Eq. 4 yields

$$\begin{aligned} \rho_{i+1}(h, t) &= \rho_i(h) m_i(h, t), \\ \nabla_\theta J_1(h_0) &= \sum_{i=1}^{B} \sum_{h \in \mathcal{H}_{i-1}} \rho_i(h) \sum_{t \in \mathcal{V}} \big[1 + V_i(h, t)\big] \nabla_\theta m_i(h, t). \end{aligned} \tag{5}$$

This identity holds wherever the overlaps are differentiable. Each candidate's credit combines prefix reachability $\rho_i(h)$ with the expected return $1 + V_i(h, t)$ conditional on accepting it. In particular, differentiating $-\sum_{i,h,t} \mathrm{sg}[\rho_i(h)(1 + V_i(h, t))] m_i(h, t)$ gives exactly $-\nabla_\theta J_1(h_0)$. Detaching these weights is valid because the decomposition already includes the derivatives of draft decisions at every position in the complete prefix tree. Appendix D gives the full expansion.

4.2 Practical Value Estimation and Training

A compact candidate graph. For the DSpark drafter with a Markov head, a single parallel decoder pass on h_0 produces base logits $b_{1:B}$. Let $g_\theta(t)$ denote the logit correction produced by the Markov head from predecessor token t, and let s_0 be the final token of h_0. We construct a sampled draft path sequentially: at position i, compute $q_i^{\mathrm{path}} = \mathrm{softmax}(b_i + g_\theta(s_{i-1}))$, retain its top-$K$ candidates C_i, and sample s_i from q_i^{path} renormalized over C_i. Within each block, the resulting candidate sets remain fixed across branches, forming the compact graph in Figure 1(b).

For $i < B$ and each candidate $t \in C_i$, we evaluate the target's next-token distribution at $(h_0, s_{<i}, t)$. The corresponding draft distribution is $\mathrm{softmax}(b_{i+1} + g_\theta(t))$. For $u \in C_{i+1}$, let $M_{i+1}(t, u)$ be the minimum of the probabilities assigned to u by these two distributions. Backward dynamic programming gives

$$\widehat{V}_i(t) = \sum_{u \in C_{i+1}} M_{i+1}(t, u)[1 + \widehat{V}_{i+1}(u)], \qquad \widehat{V}_B(t) = 0. \tag{6}$$

The DP uses $O(BK^2)$ scalar operations. Target evaluation covers $(B-1)K$ candidate nodes after a shared context prefill. The approximation restricts continuations to top-K candidates and reuses sampled earlier histories: $\widehat{V}_{i+1}(u)$ is computed under $(h_0, s_{<i+1}, u)$ and shared across predecessor candidates t. Thus, t affects the next-step overlap but is not retained in the histories used for subsequent value estimates. Top-K renormalization is used only to sample s_i; all overlaps use the original full-vocabulary probabilities. No separate value network is trained.

Value-corrected distribution matching. We use DSpark's full-vocabulary L1 loss, equal to twice the total-variation (TV) distance, as our base (Cheng et al., 2026). TV-based distillation is also studied in f-Distill (Wen et al., 2023) and DistillSpec (Zhou et al., 2024). We augment this loss with a candidate-specific continuation reward. At sampled history $\bar{h}_{i-1} = (h_0, s_{<i})$, both terms use the target and draft distributions conditioned on this history, abbreviated p_i and q_i. The context prefill and previously evaluated candidate nodes supply these target distributions without additional target

evaluations. With continuation coefficient $\beta \geq 0$, the training objective is

$$\ell_i = \|p_i - q_i\|_1 - 2\beta \sum_{t \in C_i} \min\{p_i(t), q_i(t)\}\, \mathrm{sg}[\widehat{V}_i(t)], \qquad \mathcal{L}_{\mathrm{align}} = \frac{\sum_{n,i} \omega_{n,i} \ell_{n,i}}{\sum_{n,i} \omega_{n,i}}. \tag{7}$$

Here n indexes training blocks, and $\omega_{n,i}$ is the valid-position mask multiplied by $e^{-(i-1)/\gamma}$, with $\gamma > 0$. Sampled tokens and candidate sets are treated as fixed during differentiation. Since $\|p_i - q_i\|_1 = 2 - 2\sum_{t \in \mathcal{V}} \min\{p_i(t), q_i(t)\}$, the local loss ℓ_i assigns credit per unit accepted probability mass of $1 + \beta \widehat{V}_i(t)$ within C_i and 1 outside it, up to a common factor of 2. Thus β controls the strength of continuation credit. Sampled prefixes, truncated candidate sets, shared sampled histories, and position weights make this a practical surrogate rather than an exact implementation of Eq. 5.

5 Experiments

5.1 Experimental Setup

We study whether continuation credit improves accepted length, whether candidate-specific credit matters, and how continuation strength and candidate count affect performance. We use a frozen Qwen3-8B target and a five-layer DSpark drafter with a rank-256 Markov head and block length $B = 7$. For the main training experiments, all runs start from the same draft checkpoint and train for six epochs on the full Open-Perfectblend dataset. We use a global batch size of 32, sequence length 8,192, and Adam with a peak learning rate of 6×10^{-4}, 80 warmup updates, and cosine decay. Unless otherwise stated, we use $\beta = 0.3$, $K = 4$, detached continuation-value estimates, and position decay $\gamma = 4$.

We evaluate on BBH (Suzgun et al., 2022), MMLU-Pro (Wang et al., 2024), WinoGrande (Sakaguchi et al., 2019), and GSM8K (Cobbe et al., 2021). We report average accepted length (AL), excluding correction and bonus tokens. The main comparison uses decoding temperatures $T = 0$ and $T = 1$; ablation and sensitivity studies use $T = 1$. Relative changes are computed as $100(\mathrm{AL}_{\mathrm{method}}/\mathrm{AL}_{\mathrm{base}} - 1)$ in percent, where $\mathrm{AL}_{\mathrm{base}}$ denotes the AL of the corresponding comparison baseline.

5.2 Main Comparison

Table 1: Average accepted length (AL) of DSpark, D-PACE, and D-VOLT at decoding temperatures $T = 0$ and $T = 1$. Superscripts show relative gains over DSpark; bold marks the best AL for each benchmark and temperature.

	Temperature $T = 0$			Temperature $T = 1$		
Benchmark	DSpark	D-PACE	D-VOLT	DSpark	D-PACE	D-VOLT
BBH	3.3918	3.4275 $\nearrow 1.05\%$	**3.5539** $\nearrow 4.78\%$	3.7714	3.8547 $\nearrow 2.21\%$	**4.0199** $\nearrow 6.59\%$
MMLU-Pro	4.6874	4.7896 $\nearrow 2.18\%$	**5.0990** $\nearrow 8.78\%$	3.9882	4.0426 $\nearrow 1.36\%$	**4.2833** $\nearrow 7.40\%$
WinoGrande	3.5245	3.7291 $\nearrow 5.81\%$	**3.8679** $\nearrow 9.74\%$	3.2555	3.3195 $\nearrow 1.97\%$	**3.3873** $\nearrow 4.05\%$
GSM8K	4.4614	4.5961 $\nearrow 3.02\%$	**4.7317** $\nearrow 6.06\%$	3.9875	4.1158 $\nearrow 3.22\%$	**4.3088** $\nearrow 8.06\%$

Table 1 compares D-VOLT with the DSpark baseline and D-PACE. D-VOLT achieves the highest AL on all four benchmarks at both decoding temperatures. Relative to DSpark, gains range from 4.78%–9.74% at $T = 0$ and 4.05%–8.06% at $T = 1$. Across both temperatures, gains over D-PACE range from 2.04% to 6.46%.

5.3 Mechanism Ablations

All four variants use the same sampled-prefix training setup and differ only in the continuation reward. *No value* removes the continuation reward by setting $\beta = 0$. At each sampled prefix and position, write $m_i(t) = \min\{p_i(t), q_i(t)\}$. *Shared value* replaces each $\widehat{V}_i(t)$ in the continuation reward with the detached mean $\bar{V}_i^{\mathrm{acc}} = \mathrm{sg}[(\sum_{t \in C_i} m_i(t)\widehat{V}_i(t))/(\sum_{t \in C_i} m_i(t))]$. *Candidate value*

uses each candidate's own estimate $\widehat{V}_i(t)$, as in D-VOLT. *Shuffled value* permutes these estimates among candidates at the same prefix and position. At $K = 1$, the shared, candidate-specific, and shuffled variants coincide, so distinguishing their credit assignments requires $K > 1$.

Table 2: Effect of continuation credit on average accepted length (AL) at $K = 4$ and $T = 1$. We use $\beta = 0.3$ except for No value, which sets $\beta = 0$; all other training settings are shared. The best result on each benchmark is bold.

Variant	BBH	MMLU-Pro	WinoGrande	GSM8K
No value	3.8974	4.1831	3.2962	4.2256
Shared value	3.9014	4.2092	3.3145	4.2329
Candidate value (D-VOLT)	**4.0199**	**4.2833**	**3.3873**	**4.3088**
Shuffled value	3.8702	4.1788	3.2748	4.2195

When evaluated with the same distributions, candidate sets, and value estimates, the shared and candidate-specific continuation rewards have identical forward values. The shared variant nevertheless assigns the same continuation weight to every retained candidate during differentiation. Shuffling tests candidate–value correspondence, but may also change the mean value weighted by accepted probability mass. In Table 2, Candidate value improves AL over Shared value by 1.76%–3.04% across the four benchmarks.

5.4 Hyperparameter Sensitivity

Table 3: Average accepted length (AL) of D-VOLT under different continuation strengths β and candidate counts K at $T = 1$. Panel (a) varies β with $K = 4$; panel (b) varies K with $\beta = 0.3$. The best result for each benchmark within each panel is bold.

Setting	BBH	MMLU-Pro	WinoGrande	GSM8K
(a) Continuation strength ($K = 4$)				
$\beta = 0.1$	3.9848	4.0743	3.3548	4.2967
$\beta = 0.2$	4.0005	4.1208	3.3656	4.3067
$\beta = 0.3$	**4.0199**	**4.2833**	**3.3873**	**4.3088**
$\beta = 0.4$	4.0021	4.1856	3.3468	4.1487
$\beta = 0.6$	3.9767	4.1472	3.3380	4.1722
(b) Candidate count ($\beta = 0.3$)				
$K = 1$	3.9161	4.2499	3.3677	4.2431
$K = 2$	3.9243	4.1964	3.3780	4.2301
$K = 4$	**4.0199**	**4.2833**	**3.3873**	**4.3088**
$K = 8$	3.9720	4.2116	3.2636	4.2501

At $T = 1$, we sweep $\beta \in \{0.1, 0.2, 0.3, 0.4, 0.6\}$ with $K = 4$, and $K \in \{1, 2, 4, 8\}$ with $\beta = 0.3$. We select the default configuration $\beta = 0.3$, $K = 4$ using a held-out GSM8K validation set disjoint from the reported evaluation data, and use the same defaults across benchmarks. Table 3 shows that this configuration achieves the highest AL on all four benchmarks in both sweeps.

5.5 Inference and Training Cost

Inference cost. D-VOLT preserves the deployed drafter architecture, frozen target model, and standard verification procedure. Candidate-graph construction and continuation-value estimation are used only during training. Under standard verification, each block advances $X+1$ output tokens, including the correction or bonus token, apart from end-of-sequence truncation. At fixed per-block drafting and verification costs, throughput therefore scales with $\mathrm{AL} + 1$, so relative AL gains do not directly equal relative throughput gains.

Training cost. The additional training cost comes from estimating token-conditional continuation values. For a block of length B with K retained candidates per position, target evaluation covers

$(B-1)K$ candidate nodes after a shared context prefill, while backward DP requires $O(BK^2)$ scalar operations. For our default $B = 7$ and $K = 4$, this corresponds to 24 candidate nodes per block.

We report training-step times from a separate earlier comparison, using $K = 4$ and $\beta = 0.3$ for D-VOLT. Both runs use 16 GPUs across two nodes, global batch size 256, microbatch size 4, sequence length 8,192, and matched initialization and training data. We compare steps 101–762 after excluding the first 100 startup steps. The DSpark baseline runs a diagnostic value probe every 50 steps; excluding these 13 scheduled steps from both runs leaves 649 matched training steps. Excluding checkpoint-saving time, mean step time is 4.141 s for DSpark and 20.916 s for D-VOLT, a $5.05\times$ increase. Including the diagnostic steps gives 6.130 s versus 20.921 s ($3.41\times$), reflecting the baseline's diagnostic overhead.

Across the 649 retained steps, mean forward time increases from 2.741 s to 19.576 s, while mean backward time remains similar (1.385 s versus 1.328 s). These measurements compare complete training configurations with different software versions and supervision paths at $K = 4$.

6 Related Work

Speculative decoding. Speculative decoding verifies multiple draft tokens in parallel and uses rejection sampling to preserve the target model's output distribution (Leviathan et al., 2023; Chen et al., 2023). EAGLE uses autoregressive feature prediction for drafting (Li et al., 2024a), while EAGLE-2 constructs dynamic draft trees using context-dependent draft confidence (Li et al., 2024b). DFlash uses target-conditioned block diffusion for parallel drafting (Chen et al., 2026); DSpark combines a parallel backbone with a lightweight sequential Markov head (Cheng et al., 2026).

Training speculative drafters. DistillSpec shows that both draft-generated data and the distillation divergence affect draft–target alignment (Zhou et al., 2024). Online Speculative Decoding adapts the drafter to observed queries (Liu et al., 2023). EAGLE-3's training-time test feeds the drafter's own intermediate outputs back into training (Li et al., 2025). Draft-OPD and Aurora incorporate target feedback on both accepted and rejected draft proposals (Lei et al., 2026; Wang et al., 2026). AdaSPEC uses a reference model to filter tokens during distillation (Hu et al., 2025), while LK Losses uses negative log acceptance probability or an adaptive KL/TV mixture (Samarin et al., 2026).

Training for accepted length. DSpark uses exponentially decaying position weights (Cheng et al., 2026), whereas PARD-2 weights token losses by the product of target confidences along the prefix (An et al., 2026). D-PACE derives adaptive position weights from a draft-confidence surrogate of expected accepted length; without smoothing, its weighted loss has the same gradient as the negative surrogate (Wu et al., 2026). AngelSpec optimizes a sum of cumulative products of depth-wise target–draft overlaps on training histories (Liu et al., 2026). VSD maximizes the expected log probability of accepting an entire draft path, with a path-level KL penalty (Zou et al., 2026). D-VOLT assigns candidate-specific credit using token-conditional continuation values derived over complete prefix histories.

7 Conclusion

We introduce D-VOLT, a method for training speculative drafters using token-conditional continuation values. Our analysis shows that a detached continuation weight shared across candidate tokens at the same prefix and position can favor higher one-step acceptance probability while decreasing expected accepted length, even when the weight uses the exact mean continuation value conditional on acceptance. We derive a Bellman recursion over complete prefix histories and an exact gradient decomposition of expected accepted length. Our practical objective combines full-vocabulary total-variation matching with a candidate-specific continuation reward, while preserving the deployed drafter architecture and standard verification procedure. Across four benchmarks at decoding temperatures $T = 0$ and $T = 1$, D-VOLT improves average accepted length by 4.05%–9.74% over the DSpark baseline and by 2.04%–6.46% over D-PACE.

AI USE STATEMENT

The core research idea originated with the authors. ChatGPT assisted with manuscript organization and language polishing. The authors retain responsibility for all scientific decisions, verification of results, and 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/2605.08632v1.

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/2302.01318v1.

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/2602.06036v2.

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/2607.05147v1.

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/2110.14168v2.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500830.

Yuezhou Hu, Jiaxin Guo, Xinyu Feng, and Tuo Zhao. AdaSPEC: Selective knowledge distillation for efficient speculative decoders. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=zNLlglSOwD.

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/2605.29343v2.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

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/2503.01840v3.

Hong Liu, Rui Cen, Junhan Shi, Guangshuo Qin, Jiebin Zhang, Tianyu Liu, Runzhi Fan, Guoliang Zhao, Ruobing Xie, Kai Zhang, Song Liu, Guanghua Yu, and Jianchen Zhu. AngelSpec: Towards real-world high performance inference with speculative decoding. *arXiv preprint arXiv:2607.25852*, 2026. URL https://arxiv.org/abs/2607.25852v1.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Alvin Cheung, Zhijie Deng, Ion Stoica, and Hao Zhang. Online speculative decoding. *arXiv preprint arXiv:2310.07177*, 2023. URL https://arxiv.org/abs/2310.07177.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019. URL https://arxiv.org/abs/1907.10641v2.

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/2602.23881v2.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022. URL https://arxiv.org/abs/2210.09261v1.

Junxiong Wang, Fengxiang Bie, Jisen Li, Zhongzhu Zhou, Zelei Shao, Yubo Wang, Yinghui Liu, Qingyang Wu, Avner May, Sri Yanamandra, Ce Zhang, Tri Dao, Percy Liang, Ben Athiwaratkun, Shuaiwen Leon Song, Chenfeng Xu, and Xiaoxia Wu. When RL meets adaptive speculative training: A unified training-serving system. *arXiv preprint arXiv:2602.06932*, 2026. URL https://arxiv.org/abs/2602.06932v5.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024. URL https://arxiv.org/abs/2406.01574v6.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10817–10834. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.605. URL https://aclanthology.org/2023.acl-long.605/.

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/2605.18810v1.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388v1.

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

Xiandong Zou, Jianshu Li, Jing Huang, and Pan Zhou. Variational speculative decoding: Rethinking draft training from token likelihood to sequence acceptance. *arXiv preprint arXiv:2602.05774*, 2026. URL https://arxiv.org/abs/2602.05774v6.

A CONTINUATION-VALUE DIAGNOSTIC

A.1 COLLECTION AND CONDITIONAL MONTE CARLO ESTIMATION

The diagnostic uses a five-layer DSpark drafter with a rank-256 Markov correction and a draft attention window of 4,096 tokens. The target is Qwen3-8B. We draw 100 prompts without replacement from a fixed five-shot GSM8K evaluation suite using seed 20260914, disable thinking, and cap each completion at 2,048 tokens. Drafting and stochastic verification use temperature 1, without top-p or top-k sampling truncation. Every proposed block contributes its seven prefix–position records, including prefixes whose earlier proposals may subsequently be rejected. We exclude initialization and warm-up requests. These records are therefore not weighted by the probability that verification reaches the corresponding position, and their average value is not a block-level acceptance metric.

This diagnostic uses zero-based position IDs: position $i \in \{0, \dots, 6\}$ corresponds to theoretical position $i + 1$. Let $H_i = 6 - i$ be its remaining horizon. For each recorded prefix h, position i, and candidate t, draw $M = 16$ suffix trajectories $u^{(m)}_{1:H_i}$ from the full draft continuation distribution. The candidate replaces only position i; all preceding draft tokens remain fixed. Let $p^{(m)}_r$ and $q^{(m)}_r$ be the probabilities assigned to $u^{(m)}_r$ by the target and draft under that branch's history. The target conditions on the complete prefix $(h, t, u^{(m)}_{<r})$, and the draft's Markov predecessor is the token actually sampled immediately before $u^{(m)}_r$. Define

$$\alpha^{(m)}_r = \min\left\{1, \frac{p^{(m)}_r}{q^{(m)}_r}\right\}, \qquad Y_m = \sum_{\ell=1}^{H_i} \prod_{r=1}^{\ell} \alpha^{(m)}_r, \qquad \widehat{V}_i(h,t) = \frac{1}{M} \sum_{m=1}^{M} Y_m. \tag{8}$$

The products integrate out acceptance coins conditional on the sampled suffix. Thus $Y_m \in [0, H_i]$ has the expected accepted suffix length as its mean. Repeated trajectories retain their multiplicity; batching only reuses target computation. The estimator retains full branch histories, unlike the compact top-K graph with shared sampled earlier histories used for training.

A.2 DISPERSION, AGGREGATION, AND UNCERTAINTY

For the $K = 16$ candidates in one prefix–position record, define

$$\overline{V}_i(h) = \frac{1}{K} \sum_{t \in C_i(h)} \widehat{V}_i(h,t), \qquad s^2_i(h) = \frac{1}{K} \sum_{t \in C_i(h)} (\widehat{V}_i(h,t) - \overline{V}_i(h))^2. \tag{9}$$

The scatter plots place each record's candidates at $x = \overline{V}_i(h)$ and $y = \widehat{V}_i(h,t)$, so their mean squared distance in the vertical direction from $y = x$ is exactly $s^2_i(h)$. The full-size view in Figure 4 uses the same points as the compact main-text figure. No records are filtered by value or probability.

For any candidate weights $a_t \geq 0$ summing to one within the recorded set, we compute

$$\mu_a = \sum_{t \in C_i(h)} a_t \widehat{V}_i(h,t), \qquad s^2_a = \sum_{t \in C_i(h)} a_t \big(\widehat{V}_i(h,t) - \mu_a\big)^2. \tag{10}$$

Equal weights use $a_t = 1/16$; draft weights use $a_t = q_i(t \mid h) / \sum_{u \in C_i(h)} q_i(u \mid h)$; overlap weights replace q_i with $m_i = \min(p_i, q_i)$. This normalization describes the recorded support without changing model probabilities. All standard deviations use the population convention, with no Bessel correction. The reported mean standard deviation averages s_a across records, rather than taking the square root of their mean variance.

We resample the 100 requests with replacement for 2,000 bootstrap replicates (seed 20260914), retaining all records of each sampled request. Percentile intervals for record-weighted means preserve within-request dependence but do not remove MC noise. Across the 20,280 nonterminal records, the equal-weight mean standard deviation is 0.937 tokens (95% interval $[0.918, 0.955]$). Figure 2 plots all records at positions 0 and 3 without subsampling or horizontal jitter, using the same records as the summary statistics.

After dividing by the remaining horizon $H_i = 6 - i$, mean standard deviations range from 0.245 to 0.348 across positions 0–5. A candidate value range exceeding half of H_i occurs in 88.3% of

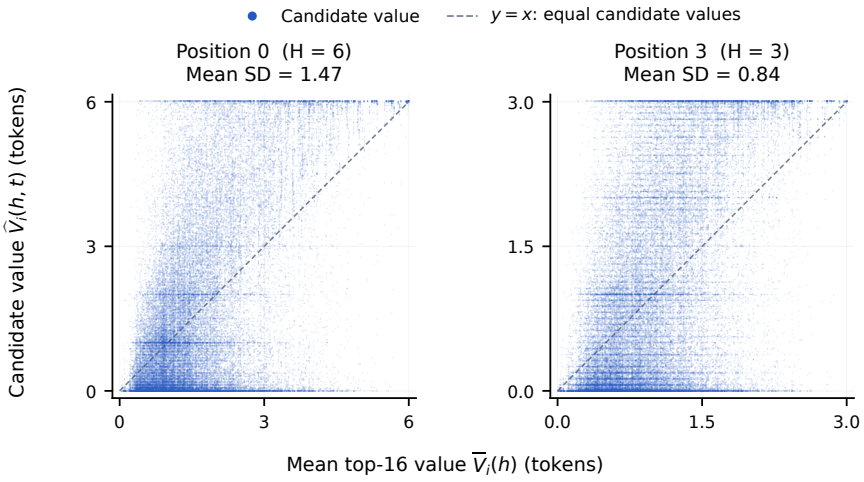

Figure 4: Full-size view of candidate-value variation. All 3,380 prefixes at each of positions 0 and 3 contribute their 16 candidate values, without subsampling or horizontal jitter. Values count suffix tokens only. Coincident points overlap; transparency reveals density.

Position	H_i	Mean s_i^2	Mean s_i	Mean s_i/H_i	Mean s_{overlap}
0	6	2.552	1.472	0.245	0.152
1	5	1.873	1.274	0.255	0.126
2	4	1.289	1.070	0.268	0.105
3	3	0.786	0.843	0.281	0.078
4	2	0.402	0.612	0.306	0.060
5	1	0.126	0.348	0.348	0.041
6	0	0	0	—	0

Table 4: Within-record dispersion of estimated top-16 continuation values. Each position contains 3,380 records. Variance is in squared tokens; unnormalized standard deviations are in tokens. All entries are descriptive statistics of the 16-draw MC estimates.

nonterminal records. The mean draft-weighted standard deviation is 0.180 tokens, compared with 0.937 under equal weights. Low-probability candidates can therefore contribute substantially to equal-weight dispersion.

The stored data contain candidate means, not individual trajectory returns. Consequently, an empirical standard error for each value or a split-sample correction of the variance cannot be recovered. A conservative bounded-MC check is still possible. Under independent draws within each candidate and a fixed numerical probability evaluator, Hoeffding's inequality (Hoeffding, 1963), followed by a union bound over the K candidates, gives

$$\Pr\left[\max_{t\in C_i(h)} |\widehat{V}_i(h,t) - V_i(h,t)| > \epsilon_i\right] \le \delta, \qquad \epsilon_i = H_i\sqrt{\frac{\log(2K/\delta)}{2M}}. \tag{11}$$

No independence across candidates is required for this union bound. Writing $\widehat{R}_i(h) = \max_t \widehat{V}_i(h,t) - \min_t \widehat{V}_i(h,t)$, a simultaneous lower bound on the true candidate range within that record is $\max\{0, \widehat{R}_i(h) - 2\epsilon_i\}$. For $K = M = 16$ and $\delta = 0.05$, $2\epsilon_i = 0.898712H_i$. The bound is positive for 36.1%, 40.7%, 44.7%, 51.8%, 67.0%, and 88.7% of records at positions 0 through 5, respectively. The coverage is 95% within each record, not simultaneously over all records; these fractions are not 95% lower bounds on population prevalence. The calculation concerns sampling noise under its stated assumptions and does not bound systematic errors from finite-precision model evaluation. It is a sensitivity check, not a noise-corrected estimate of the variance.

B Oracle Allocation Within and Across Prefixes

B.1 Objective and Budget

We reuse the 23,660 records and MC values from Appendix A. Each position has 3,380 records from 100 requests. Records receive equal weight, so requests with more recorded blocks contribute more; this is not a question-weighted mean. Let $\mathcal{D}_i$ contain the recorded prefixes at position i and $C_i(h)$ their top-16 candidates. The pooled accepted-mass relaxation is

$$\max_{0 \le z_{h,t} \le p_i(t|h)} \frac{1}{|\mathcal{D}_i|} \sum_{h,t} z_{h,t}(1 + \widehat{V}_i(h,t)) \quad \text{s.t.} \quad \sum_{h,t} z_{h,t} = \sum_{h,t} m_i(h,t). \tag{12}$$

Both sums range over $h \in \mathcal{D}_i$, $t \in C_i(h)$. For a fixed position, abbreviate $p_{h,t} = p_i(t \mid h)$, $q_{h,t} = q_i(t \mid h)$, $v_{h,t} = \widehat{V}_i(h,t)$, and $m_{h,t} = \min(p_{h,t}, q_{h,t})$, with $t \in C_i(h)$. The baseline retained-support return is $\overline{J}_i = |\mathcal{D}_i|^{-1} \sum_{h,t} m_{h,t}(1 + v_{h,t})$. Equation 12 fixes the position's total budget $B_i = \sum_{h,t} m_{h,t}$ and reallocates it among records and candidates. Different positions never exchange budget. The per-record comparison instead imposes $\sum_t z_{h,t} = \sum_t m_{h,t}$ separately for every h.

The probabilities retain their original full-vocabulary normalization. Only recorded candidates change; the unrecorded tail is held fixed and cancels from the difference in return and mean acceptance. Absolute returns and relative-gain denominators include only top-16 contributions. Mean recorded draft mass ranges from 99.33% to 99.92% across positions; mean recorded target mass ranges from 94.06% to 99.58%. Holding the pooled accepted mass fixed means the unit reward also cancels, so maximizing $\sum z(1+v)$ is equivalent to maximizing $\sum zv$.

B.2 Solution, Feasibility, and Verification

The pooled problem is a capped linear allocation: sort all candidates at the same position by descending v, then fill each target cap until B_i is exhausted. For any scalar λ, a dual upper bound on the total return is

$$\lambda B_i + \sum_{h,t} p_{h,t} \max\{1 + v_{h,t} - \lambda, 0\}. \tag{13}$$

Choosing the marginal allocated value gives equality. Across all seven positions, budget residuals are below 10^{-10} and primal–dual objective differences below 2×10^{-10}. An independent linear-program solver agrees within 2.1×10^{-8} in total return. The pooled optimum is at least the per-record optimum, which is at least the baseline. At the last position, $v = 0$ and all three returns equal the fixed acceptance budget.

Accepted-mass feasibility alone does not guarantee a normalized proposal with unchanged tail. We therefore also construct a feasible lower bound. For each record, let $Q_h = \sum_t q_{h,t}$ and choose $a_h = \arg\min_t p_{h,t}$. Keeping the pooled budget $\sum_{h,t} z_{h,t} = B_i$ fixed, set $z_{h,a_h} = p_{h,a_h}$ and allocate the remaining mass by value subject to target caps and $\sum_t z_{h,t} \le Q_h$. Then set

$$q'_{h,t} = z_{h,t} + \mathbf{1}[t = a_h] \left(Q_h - \sum_u z_{h,u} \right). \tag{14}$$

The saturated candidate absorbs surplus proposal mass. With the unchanged vocabulary tail, q'_h is normalized and $\min(p_{h,t}, q'_{h,t}) = z_{h,t}$. We verify both identities numerically to within 5×10^{-16}. This construction is a feasible lower bound, not an exact optimizer of the proposal-constrained problem. Its aggregate gain is 5.073%, versus the relaxation's 5.096%; the largest per-position difference in mean return is below 0.001 token. The gap is thus not explained by the accepted-mass relaxation alone. These per-record proposals are not claimed to be realizable by one shared finite-capacity network.

B.3 Results and Interpretation

The mean local return across all records changes from 2.573066 to 2.704196, a 5.096% increase. The per-record oracle yields 2.662871 (+3.490%). The per-record oracle isolates within-prefix candidate allocation, whereas the pooled oracle additionally allows reallocation across prefixes.

Position	Overlap	Current	Per-record	Pooled	Gain (95% CI)
0	0.8830	4.4231	4.5842	4.6251	4.57% [4.05, 5.10]
1	0.8587	3.7841	3.9205	3.9767	5.09% [4.57, 5.64]
2	0.8372	3.1827	3.2992	3.3562	5.45% [4.92, 5.99]
3	0.8241	2.5906	2.6835	2.7334	5.51% [4.96, 6.13]
4	0.7860	1.9663	2.0385	2.0913	6.36% [5.80, 6.93]
5	0.7522	1.3526	1.4021	1.4346	6.06% [5.55, 6.60]
6	0.7121	0.7121	0.7121	0.7121	0.00% [0.00, 0.00]

Table 5: Oracle allocation with unchanged mean acceptance at each position. Returns include the current draft token and exclude correction and bonus tokens; only top-16 contributions are included. Gain compares the pooled oracle with the current return. Interval endpoints are in percent.

We resample the 100 requests with replacement for 2,000 replicates (seed 20260915), preserving all records from each sampled request. Crucially, each replicate recomputes its position-wise budget and re-solves the pooled allocation. The percentile intervals in Table 5 describe request-sampling uncertainty for this record-weighted diagnostic. They do not correct the optimism from selecting and evaluating allocations with the same finite-MC values. Independent suffix draws would be needed for that evaluation.

The analysis holds baseline continuations and recorded prefixes fixed. Changing a drafter throughout a block would also change suffix values and the distribution of reached histories. Moreover, these records include prefixes that verification need not reach. Neither sums nor averages of their local returns are measurements of whole-block accepted length. The oracle describes allocation headroom under the stated relaxation; it is not a measured speedup or a fixed-capacity model optimum.

C Details of the Shared-Weight Counterexample

Consider three tokens with fixed continuation values $V = (1, 4, 0)$ and

$$p = (0.50, 0.40, 0.10), \qquad q = (0.30, 0.40, 0.30) \longrightarrow q' = (0.40, 0.35, 0.25). \tag{15}$$

Accepted masses change from $(0.30, 0.40, 0.10)$ to $(0.40, 0.35, 0.10)$. Total overlap rises from 0.80 to 0.85, while L1 decreases from 0.40 to 0.30 and target-weighted cross-entropy decreases from 1.0889 to 1.0167 nats. Nevertheless, J falls from 2.70 to 2.65. Figure 3 follows the update candidate by candidate; the distributions and continuation values are all feasible.

For Equation 15, interpolate between the two draft distributions as $q(a) = (0.30 + 0.10a,\ 0.40 - 0.05a,\ 0.30 - 0.05a)$, $a \in [0, 1]$. With p and the downstream values fixed,

$$A(a) = 0.80 + 0.05a, \qquad J(a) = 2.70 - 0.05a, \qquad \|p - q(a)\|_1 = 0.40 - 0.10a. \tag{16}$$

The target-weighted cross-entropy $-\sum_t p_t \log q_t(a)$ is convex in a, and its derivative at $a = 1$ is $-67/1400 < 0$. Its derivative is therefore negative throughout the interval. Thus both local losses decrease along a feasible direction on which accepted length decreases. With at least four remaining positions, matching or disjoint deterministic distributions can make the three branches accept one, four, and zero tokens. Strictly positive softmax distributions can approximate this construction while preserving the strict inequalities.

The example concerns a shared, positive, detached weight multiplying a local loss. Such a weight preserves the loss's local preferences but cannot supply the accepted-token composition term in Equation 3. Differentiating through every dependency of an exact $\widetilde{V}$ would preserve that identity. The mismatch disappears when candidate values agree or feasible updates cannot alter their relative accepted mass. In particular, D-PACE's unsmoothed weights recover its own confidence-surrogate gradient and depend on the sampled trajectory (Wu et al., 2026). This construction is not an end-to-end counterexample to that full algorithm.

D Acceptance-Value Derivation and Implementation

Full-history expectation. Fix h_0 independently of the drafter parameters. For a draft sequence $z_{1:k}$, let $h_{j-1} = (h_0, z_{<j})$. Under standard stochastic verification, the probability of proposing and accepting that particular prefix is $\prod_{j=1}^{k} m_j(h_{j-1}, z_j)$. Summing over all prefixes and using the tail-sum identity for the nonnegative integer X gives

$$J_1(h_0) = \sum_{k=1}^{B} \sum_{z_{1:k} \in \mathcal{V}^k} \prod_{j=1}^{k} m_j(h_{j-1}, z_j). \tag{17}$$

The target is frozen, but its distribution at each node retains the entire branch history. Regrouping the first edge yields Eq. 4. There is no top-K restriction, sampled-history replacement, or approximation to V in this identity. It describes accepted draft tokens in a block; the replacement or bonus token is excluded.

Exact gradient decomposition. Differentiate every product in Eq. 17. For the derivative of the edge (i, h, t), preceding factors sum to its reach probability $\rho_i(h)$. Suffix factors contribute one for accepting the current token and $V_i(h, t)$ for all longer accepted continuations. Their product gives Eq. 5, including when parameters are shared across positions and histories. All sums are finite. The equality holds away from ties $p_i(t \mid h) = q_i(t \mid h)$; at a tie, an implementation can select a consistent derivative of the minimum.

An equivalent full-tree training surrogate is

$$\mathcal{L}_{\text{exact}} = \frac{1}{2} \sum_{i,h} \text{sg}[\rho_i(h)] \left(\|p_i(\cdot \mid h) - q_i(\cdot \mid h)\|_1 - 2 \sum_{t \in \mathcal{V}} m_i(h, t)\, \text{sg}[V_i(h, t)] \right). \tag{18}$$

Its gradient is $-\nabla_\theta J_1(h_0)$. The values and reach probabilities are evaluated at the current parameters and then held fixed for the update. This equality does not require treating V as independent of the drafter: the derivatives of future decisions appear separately at their own edges. Conversely, differentiating through V again in this particular full-tree surrogate would count those contributions again. This distinction separates an exact credit decomposition from directly differentiating the root Bellman value, which also gives the exact gradient.