BAGDRAFT: NEXT-BAG-OF-TOKENS DRAFTING FOR LOSSLESS SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

The speedup of speculative decoding is fundamentally bounded by draft token throughput, yet existing drafters either incur K sequential passes (EAGLE), dilute model capacity across K disjoint heads (Medusa), or tie capacity to rigid per-slot block predictions (DFlash). We propose **BagDraft**, which reformulates speculative drafting as a next-bag-of-tokens prediction task. Crucially, the informational debt of discarding sequence order requires no parametric recovery in speculative decoding, as exact target verification guarantees lossless generation. Operating via a single forward pass with a shared logit vector, BagDraft directly models future K-token sets under a multi-hot cross-entropy objective, using *discrete anchor blocks* to ensure strict input isomorphism between training and inference. At inference, we mitigate the $K!$ permutation explosion by formalizing path probabilities via Plackett–Luce sampling without replacement; its proven prefix-monotonicity property mathematically justifies expectation-based pruning to construct compact permutation trees, verified by the target model in a single forward pass. Across 7 benchmarks on Qwen3-4B/8B, BagDraft achieves an average acceptance length of 3.976 and a $3.168\times$ wall-clock speedup on 4B ($+12.1\%$ over DFlash, $+61.0\%$ over EAGLE-3), proving that speculative drafters can surpass state-of-the-art efficiency without dedicated per-position heads or autoregressive rollout, relying solely on an offset-decayed shared distribution.

1 INTRODUCTION

1.1 THE SPEEDUP CEILING IS SET BY WHAT A SINGLE DRAFTER FORWARD PRODUCES

Autoregressive decoding of large language models is fundamentally memory-bandwidth bound: each generated token requires reading the entire parameter set from memory, leaving on-chip compute underutilized (Leviathan et al., 2023; Chen et al., 2023). Speculative decoding mitigates this bottleneck by having the target model verify multiple candidate tokens in parallel within a single forward pass, raising arithmetic intensity without altering the output distribution (Xia et al., 2024). The resulting wall-clock gain follows:

$$\text{speedup} \;=\; \frac{A \cdot F_{\text{base}}}{F + O}, \tag{1}$$

where A is the acceptance length, F_{base} and F denote target forward latencies for baseline and verification steps, and O accounts for drafting and tree scheduling. Within a bounded tree budget verification stays sub-linear ($F \approx F_{\text{base}}$), so the speedup ceiling is set primarily by how many high-confidence candidates a single drafter forward can emit; exceeding the budget inflates $F > F_{\text{base}}$ and necessitates principled tree pruning (§3.4). Indeed, modern parallel drafters already accept nearly all generated tokens in a large fraction of cycles, leaving verification compute under-saturated and stranded speedup unrealized (Wu, 2026).[1]

Producing K future candidates per speculative step poses a structural dilemma for the three dominant drafting paradigms: (1) *sequential autoregressive rollout* (EAGLE; Li et al., 2024b; 2026b;

[1]We focus on the latency-critical single-request regime (batch size $= 1$); large-batch serving shifts this trade-off by saturating compute (Shi et al., 2026), while separate lines forgo external drafters via self-speculation (Sahoo et al., 2026).

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.

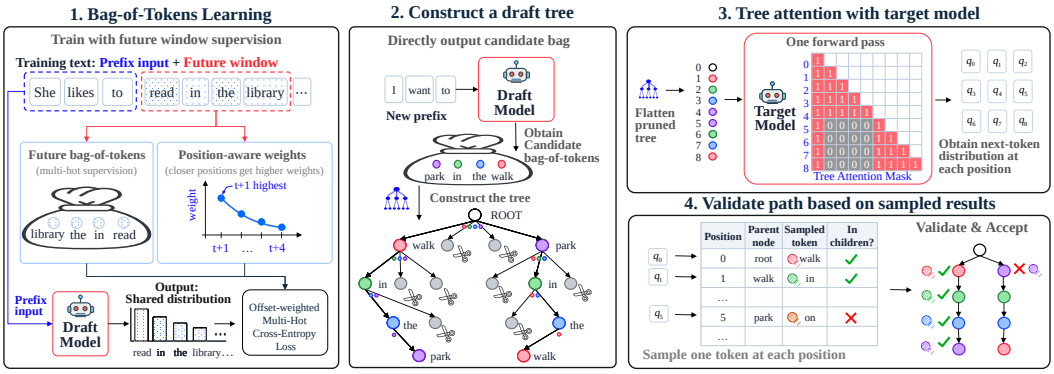


Figure 1: Overview of the BagDraft framework. **(1) Bag-of-Tokens Learning:** The draft model takes prefix text to predict an unordered bag of future tokens via a single forward pass, supervised by an offset-weighted multi-hot cross-entropy (MCE) loss with exponential position-decay weights. **(2) Construct a Draft Tree:** At inference, the drafter outputs a candidate token bag; candidate sequence trajectories are modeled via Plackett–Luce sampling without replacement, employing prefix-monotonic pruning to construct a compact permutation tree. **(3) Tree Attention with Target Model:** The flattened tree is verified in a single forward pass of the target model under a tree attention mask that restricts receptive fields strictly to ancestor paths. **(4) Validate Path Based on Sampled Results:** Tokens are independently sampled per evaluated node via Gumbel-max draws to match candidate edges, committing the longest accepted chain alongside a bonus token to guarantee exact distribution losslessness.

Zhang et al., 2025), whose K serial passes scale overhead O linearly with K; (2) *parallel multi-head amortization* (Medusa; Cai et al., 2024; Hydra; Ankner et al., 2024; MTP; Gloeckle et al., 2024; Liu et al., 2024; Gerontopoulos et al., 2026), which predicts K positions in one pass but dilutes capacity across disjoint heads, decaying sharply at deeper offsets; and (3) *block-diffusion denoising* (DFlash; Chen et al., 2026; Ringel & Romano, 2026; Zhang et al., 2026a;c; Arriola et al., 2025), which fills block slots in one pass but ties capacity to rigid per-slot marginals decoupled from selected trajectory tokens (Li et al., 2026a). Despite diverse architectures, these paradigms share the *per-position distribution hypothesis*: the drafter must explicitly model distributions at designated offsets k, either introducing serial execution dependencies or diluting representation capacity, capping end-to-end efficiency far below its potential. Orthogonal works optimize tree verification (Miao et al., 2024; Wang et al., 2025; Chen et al., 2024), cost-aware node budgets (Zhang et al., 2026b), or generalized acceptance rules (Sun et al., 2025; Weng et al., 2026), compounding with rather than replacing drafter candidate quality.

1.2 Order need not be paid for by the drafter's parameters

Discarding positional constraints has precedent in pre-training efficiency. Token-Superposition Training (TST; Peng et al., 2026) merges consecutive tokens into a bag and predicts unordered next-bag membership via multi-hot cross-entropy (MCE), alongside related formulations predicting relative token order (Zuhri et al., 2025), future summaries (Mahajan et al., 2026), or non-consecutive position sets (Gat et al., 2025). In text generation, however, lost sequence order is an informational debt that must be repaid by model parameters: sampling from a bag yields a superposition of tokens rather than fluent text, compelling TST to append an expensive token-level recovery phase to relearn sequential order.

Our core insight is that speculative decoding fundamentally changes who bears this debt. Because the drafter acts purely as a proposer while target verification is mathematically exact, ordering errors incur only marginal latency rather than distribution distortion or generation errors. Consequently, the bag objective can be preserved end to end, offloading sequence reconstruction entirely to inference-time permutation search and exact verification. This cost transfer is grounded in any-order generation theory, which shows that token difficulty depends strongly on reveal order (Kim et al., 2025) and that autoregressive models generalized to arbitrary token groups match diffusion models (Du et al.,

2026). Moving sequence order from model weights into inference search raises two foundational challenges:

First, *form isomorphism between training and inference*. A drafter should be trained in the exact format in which it is invoked (Zhang et al., 2025; Li et al., 2026b; Gu et al., 2026): at inference it takes a single query (the last committed token) with target context. Since causal-sequence training leaks preceding draft states while one-position-per-sequence training wastes data, we adopt *discrete anchor blocks*, packing hundreds of anchors under a block-sparse mask that enforces strictly diagonal draft-to-draft attention, so one forward pass realizes hundreds of inference-isomorphic single-step episodes (§3.3).

Second, *taming the permutation search space*. A pool of n tokens admits $P(n, K)$ permutations ($> 6.6 \times 10^5$ for $n = 12, K = 6$), far exceeding single-forward budgets, and without per-position conditionals path scoring is non-trivial. We formalize candidate extraction as Plackett–Luce sampling without replacement (Kool et al., 2019) and prove its path probability is *prefix-monotone*, which legitimizes expectation-based pruning (Wang et al., 2025) into compact permutation trees verified losslessly in one target forward (Miao et al., 2024). Local token repetitions outside without-replacement scoring are handled by supplementary unranked leaves (§3.4).

1.3 CONTRIBUTIONS

We present BAGDRAFT, a speculative decoding framework that fully decouples the prediction of unordered candidates from the reconstruction of their order. Our contributions are:

A next-bag-of-tokens formulation of drafting. We depart from the per-position modeling dogma, showing that a drafter without explicit position-specific parameterization—exposing only a single shared logit vector with an offset-decayed ranking prior—can emit a high-quality set of future tokens in one step and eliminates the secondary recovery training phase mandated by prior set-level paradigms.

Discrete anchor blocks with train-inference isomorphism. We introduce a block-sparse masking scheme that simulates hundreds of mutually invisible, independent single-step drafting episodes within a single sequence. This removes temporal representation leakage while maximizing training data throughput, complementing works that reshape supervision signals (Zhang et al., 2025; Gu et al., 2026).

Prefix-monotone Plackett–Luce pruning and lossless verification. To resolve the permutation explosion, we formalize candidate trajectory probabilities via Plackett–Luce sampling without replacement, prove its prefix monotonicity, and mathematically justify expectation-based tree pruning over a shared distribution. Paired with verification-side optimizations, the permutation tree is verified exactly in a single target forward pass across arbitrary sampling temperatures (Sun et al., 2023; Khisti et al., 2025; Sun et al., 2025; Weng et al., 2026).

Consistent empirical gains across model scales. Across seven benchmarks on Qwen3-4B and Qwen3-8B (Yang et al., 2025; Chen et al., 2021; Austin et al., 2021; Jain et al., 2025; Zheng et al., 2023; Li et al., 2024a; Dubois et al., 2023) at $T = 1$, BAGDRAFT improves both acceptance length and end-to-end speedup over the state-of-the-art DFlash and EAGLE-3 drafters, sweeping all benchmarks on 4B and preserving the acceptance advantage on 8B, where profiling reveals that per-node verification cost scales near-linearly with target parameters.

2 RELATED WORK

Speculative Decoding and Tree Verification. Speculative decoding (Leviathan et al., 2023; Chen et al., 2023; Xia et al., 2024) accelerates autoregressive generation via parallel target verification, where expanding candidate chains into trees (Miao et al., 2024; Wang et al., 2025; Chen et al., 2024) maximizes the expected acceptance length A. Two recent themes directly motivate our design: (i) resolving train-inference distribution shifts (Zhang et al., 2025; Li et al., 2026b), which our discrete anchor blocks address for non-sequential bag objectives; and (ii) hardware-aware budget

pricing (Zhang et al., 2026b), balancing A against verification overhead. Crucially, while existing draft trees branch on per-depth conditional distributions, BAGDRAFT constructs trees from a *single shared distribution*.

Positional Spectrum in Drafters. Existing drafters span a continuum of positional granularity: (1) full autoregressive conditionals (EAGLE; Li et al., 2024b; 2026b), providing rich context at the cost of d serial forward passes; (2) parallel marginal heads (Medusa; Cai et al., 2024; MTP; Gloeckle et al., 2024; Liu et al., 2024; Gerontopoulos et al., 2026), which predict K offsets in one pass but dilute representation capacity across disjoint heads; and (3) per-slot block distributions (DFlash; Chen et al., 2026; Arriola et al., 2025; Nie et al., 2026), extended via best-first trees (DDTree; Ringel & Romano, 2026), layer fusion (DFlare; Zhang et al., 2026a), or discrete diffusion (Set Block Decoding; Gat et al., 2025). Unlike these methods that rely on slot-specific confidences, BAGDRAFT operates at the extreme end: emitting a single shared distribution over an unordered set, requiring a novel path-scoring formulation.

Next-Bag Objectives and Multi-Hot Cross-Entropy. Predicting future token sets was introduced for pre-training efficiency via multi-hot cross-entropy (TST; Peng et al., 2026), alongside objectives predicting relative orders (Zuhri et al., 2025) or summaries (Mahajan et al., 2026). In generative pre-training, discarding sequence order incurs an informational debt that must be repaid by model parameters, necessitating a costly recovery phase. In speculative drafting, however, exact target verification guarantees lossless generation, meaning ordering errors only incur marginal latency. Consequently, BAGDRAFT completely bypasses recovery training. Moreover, TST's uniform MCE is a $\gamma \to \infty$ special case of our weighted MCE, where the decay parameter γ explicitly dictates depth-wise capacity allocation.

Sampling Without Replacement and Order Allocation. To map an unordered bag to candidate trajectories, we formalize path scores via Plackett–Luce sampling without replacement (Kool et al., 2019), proving its prefix monotonicity to theoretically justify expectation-based tree pruning. More broadly, whether sequence order belongs to training or inference has surfaced in masked diffusion (Kim et al., 2025) and any-order modeling (Du et al., 2026).

3 METHOD

BAGDRAFT decouples candidate identification from ordering: the drafter predicts an unordered bag in one forward pass, while an inference-time permutation tree reconstructs sequential order for exact target verification. Figure 1 illustrates the four stages of this pipeline, from bag-of-tokens learning to lossless tree verification.

3.1 PROBLEM SETUP

Let $q(\cdot \mid x)$ denote the target model, p_θ the drafter, and V the vocabulary size. At decoding step with committed prefix $x_{1:t}$, conventional drafters output position-specific distributions $\{p_\theta^{(k)}(\cdot \mid x_{1:t})\}_{k=1}^{K}$. Conversely, BAGDRAFT emits a single shared logit vector $z \in \mathbb{R}^V$ whose softmax $s = \mathrm{softmax}(z)$ serves as a membership distribution over the multiset:

$$\mathcal{Y} = \{\{x_{t+1}, \ldots, x_{t+K}\}\}, \tag{2}$$

retaining token multiplicity without offset indexing. At inference, candidate tokens are drawn from pool $C = \text{top-n}(s)$ with $n \geq K$.

3.2 OFFSET-WEIGHTED MULTI-HOT CROSS-ENTROPY

For logits z and token label y, the standard cross-entropy is:

$$\mathcal{L}_{\mathrm{CE}}(z, y) = -z_y + \log \sum_{i=1}^{V} \exp(z_i). \tag{3}$$

Replacing a one-hot label by multiset $\mathcal{Y}$ yields uniform multi-hot cross-entropy (Peng et al., 2026):

$$\mathcal{L}_{\mathrm{MCE}}(z, \mathcal{Y}) = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \mathcal{L}_{\mathrm{CE}}(z, y) - \log |\mathcal{Y}|. \tag{4}$$

The final term has zero gradient, so training uses the average of standard cross-entropies. While TST superposes input embeddings and requires secondary recovery training, BAGDRAFT modifies only the output target: queries receive the committed token embedding alongside target context, directly functioning as a draft proposer.

Let $\mathcal{B}$ denote supervised anchors across a global batch, with z_b the shared logit at anchor b and $y_{b,k} = x_{t_b+1+k}$ its offset-k label. We minimize:

$$\mathcal{L}_{\mathrm{bag}} = \frac{\sum_{b \in \mathcal{B}} \sum_{k=0}^{K-1} w_k m_{b,k} \mathcal{L}_{\mathrm{CE}}(z_b, y_{b,k})}{\sum_{b \in \mathcal{B}} \sum_{k=0}^{K-1} w_k m_{b,k}}, \qquad w_k = \exp(-k/\gamma), \tag{5}$$

where $m_{b,k} \in \{0, 1\}$ indicates supervision validity. The denominator aggregates globally across the batch to avoid upward reweighting of boundary anchors ($m_{b,k} = 0$). Following DFlash (Chen et al., 2026), exponential decay prioritizes early offsets that gate prefix verification. Given prefix $x_{1:t}$, define the $(k+1)$-step marginal:

$$q_k(v) = \Pr[x_{t+1+k} = v \mid x_{1:t}], \qquad k = 0, \ldots, K-1. \tag{6}$$

The marginalization in equation 6 is necessary because the drafter observes only $x_{1:t}$. Let $\tilde{w}_k = w_k / \sum_{j=0}^{K-1} w_j$ and consider anchors with a complete future window, so $m_k = 1$.

Proposition 1 (Offset weights select the fitted distribution). For independent position-specific distributions $p_k \in \Delta^{V-1}$, the unique minimizer of $\sum_{k=0}^{K-1} w_k \, \mathbb{E}_{y_k \sim q_k}[-\log p_k(y_k)]$ is $p_k^\star = q_k$ for every k, independently of $\{w_k\}$. For one shared distribution $s \in \Delta^{V-1}$, the unique minimizer of the population risk induced by equation 5 is:

$$s^\star = \sum_{k=0}^{K-1} \tilde{w}_k q_k. \tag{7}$$

Thus, a position weight only rescales an independent objective, whereas it selects the target distribution under a shared logit. At the endpoints, $\gamma \to \infty$ recovers uniform MCE, while $\gamma \to 0^+$ recovers next-token prediction. In deterministic windows ($q_k = \delta_{y_k}$), equation equation 7 becomes a weighted empirical frequency over the window. Repeated tokens consequently receive proportionally more mass, which motivates including repeated-token leaves in the inference-time permutation tree. Let $\Phi(q_0, \ldots, q_{K-1}) = \sum_k \tilde{w}_k q_k$ denote the map in equation 7.

Corollary 1 (Position is unidentifiable, but decay leaves a ranking signal). Under the setting of Proposition 1, uniform weights make Φ invariant to permutations of its arguments. For positive weights, $K \geq 2$, and every full-support $\mathcal{Q} = (q_0, \ldots, q_{K-1})$, the fiber $\Phi^{-1}(\Phi(\mathcal{Q}))$ contains a convex set with $\mathcal{Q}$ as a relative interior point and dimension $(K-1)(V-1)$. Hence no map $\Psi : \Delta^{V-1} \to (\Delta^{V-1})^K$ inverts Φ on any full-support open set. Conversely, when $q_k = \delta_{y_k}$ and the K tokens are distinct, $s^\star(y_k) = \tilde{w}_k$; for finite γ, sorting $s^\star$ in descending mass exactly recovers $(y_0, \ldots, y_{K-1})$.

This formalizes BAGDRAFT's division of labor: while a shared logit cannot reconstruct full marginals, finite decay γ embeds an ordinal ranking prior that guides inference tree search.

3.3 ANCHOR-BLOCK TRAINING ISOMORPHIC TO INFERENCE

The drafter backbone stacks five Qwen3 layers initialized from the target model's bottom five layers. We concatenate hidden states from $m = 5$ evenly spaced target layers $(\ell_1, \ldots, \ell_m)$ and compute:

$$H_{\mathrm{ctx}} = \mathrm{RMSNorm}\left(W_c[H^{(\ell_1)}; \ldots; H^{(\ell_m)}]\right). \tag{8}$$

Each draft layer uses its own key and value projections of H_{ctx} concatenated with its draft hidden states (Chen et al., 2026; Li et al., 2026b). Target embedding and LM head layers initialize the

corresponding draft modules and remain frozen, retaining vocabulary coordinates; the backbone, final norm, and context-fusion layer are trainable.

At decoding, each draft call contains one query: the last committed token. Standard causal sequence training exposes queries to earlier draft states that do not exist at decoding, whereas training one position per sequence wastes data. We therefore sample $A_{\text{anc}} = 512$ anchors $a_0 < \cdots < a_{A_{\text{anc}}-1}$ per sequence of length S (Chen et al., 2026). With query anchors $r \in \{0, \ldots, A_{\text{anc}} - 1\}$ and draft queries occupying $j \in \{S, \ldots, S + A_{\text{anc}} - 1\}$, the sparse attention mask is:

$$\text{mask}(r, j) = [(j < S) \wedge (j \leq a_r) \ \vee \ (j \geq S) \wedge (r = j - S)] \wedge v_r, \tag{9}$$

where v_r marks valid anchors and $j \leq a_r$ admits target context up to the committed anchor token. Because draft-to-draft attention is strictly diagonal ($r = j - S$), one packed forward simulates A_{anc} independent, single-query calls without leakage. Moreover, emitting a shared logit collapses the training block to $b = 1$ (reducing KV length from bA_{anc} to A_{anc}), eliminating mask tokens and intra-block bidirectional attention.

3.4 From an Unordered Bag to a Permutation Tree

Because BagDraft acts purely as a proposer, sequence order is reconstructed at inference via tree enumeration without a token-level recovery phase.

Assumption 1 (Sequential sampling without replacement). Let $C = \text{top-n}(s)$ be the candidate pool, with $s_i > 0$ for $i \in C$ and $S_C = \sum_{i \in C} s_i$. We score an ordered sequence of distinct candidates as if drawn sequentially from s restricted to C, without replacement. This assumption provides the two properties required for tree construction: its depth-one marginal agrees with s/S_C, and its path score is prefix-monotone.

Proposition 2 (Path probability). Under Assumption 1, the probability of drawing a length-d sequence $\pi = (i_1, \ldots, i_d)$ of distinct elements of C is:

$$\hat{p}(\pi) = \prod_{r=1}^{d} \frac{s_{i_r}}{S_C - \sum_{u<r} s_{i_u}}. \tag{10}$$

The probability sums to one over all such length-d sequences, and its depth-one marginal is s/S_C. Equation equation 10 is the Plackett–Luce ranking distribution on pool-normalized weights, equivalently induced by Gumbel-top-k sampling (Kool et al., 2019). The denominator uses remaining mass inside C, not $1 - \sum_{u<r} s_{i_u}$: tokens outside the pool never form tree nodes and must not penalize deeper paths (proof in Appendix A.10).

Proposition 3 (Prefix monotonicity). For any path $\pi = (i_1, \ldots, i_d)$ with $d < |C|$ and any unused $i_{d+1} \in C$, its extension $\pi' = (\pi, i_{d+1})$ satisfies $\hat{p}(\pi') \leq \hat{p}(\pi)$, strictly decreasing when $K < |C|$ (Appendix A.10).

Prefix monotonicity mathematically justifies expectation-based depthwise pruning (Wang et al., 2025). Starting at the root, we expand legal unused-token extensions and retain the top B partial paths ranked by $\log \hat{p}$ at every depth, up to depth K, where $n > K$ raises coverage without additional draft forward passes.

Because beam budget B is scarce while extra verification nodes are cheap within launch-bound regimes, we introduce three width-compensation mechanisms appended outside the beam: (1) *Pinned depth-1*: We pin the depth-one node set to all n pool candidates, independent of the beam. (2) *Linear parent widening*: Deduplicated prefixes of the top P beam paths (ranked by raw $\hat{p}$) act as parents, widened to:

$$\min(d + \Delta,\ n - d + 1) \tag{11}$$

distinct child tokens, where Δ is width offset and $n-d+1 = |C \backslash \pi|$ is remaining pool capacity. Slots are filled with unused candidates of highest shared-logit score s_i (Corollary 1). (3) *Repeated-token leaves*: To cover windows with token repetitions (which without-replacement scoring assigns zero probability), we append a bounded number R of childless leaves carrying prefix-duplicate tokens over reach-sorted parents.

Proposition 4 (Prefix-local monotonicity in P). Fix a step-start prefix, and hence s, n, B, K, and Δ. Suppose the second width-compensation mechanism has no global cap and repeated-token leaves are disabled. Let $\mathcal{T}(P)$ be the resulting tree, and let $A(\mathcal{T})$ be the number of tokens committed in that step by the verification rule below. Then $\mathcal{T}(P) \subseteq \mathcal{T}(P+1)$ and, under a coupling that shares target samples at common tree nodes:

$$A(\mathcal{T}(P)) \leq A(\mathcal{T}(P+1)) \tag{12}$$

pointwise. Hence $\mathbb{E}[A]$ is non-decreasing in P for that fixed prefix (Appendix A.10).

3.5 Lossless Verification in One Target Forward

We flatten the candidate tree into a token sequence and apply a tree-attention mask derived from ancestor relations. One target-model forward produces hidden states at all tree positions. To decide acceptance, the LM head is needed only at interior nodes; we skip the head on every childless leaf in the shared forward, projecting only interior rows.

For temperature $T > 0$, let q_j be the target distribution at node j. We draw one independent Gumbel-max sample:

$$\tilde{y}_j = \arg\max_v \left(\log q_j(v) + g_{jv}\right), \qquad g_{jv} \sim \mathrm{Gumbel}(0,1). \tag{13}$$

A child c is accepted exactly when its token equals its parent sample $\tilde{y}_{\mathrm{par}(c)}$. Because siblings carry distinct tokens, at most one child of a parent can be accepted. We commit the longest fully accepted root-to-node chain and append one bonus token from its terminal node. When rejection stops the chain at an interior node, the bonus token is its precomputed sample $\tilde{y}_j$; when the chain consumes an entire root-to-leaf path, we evaluate the LM head on that single leaf row on demand and draw its bonus token with an independent Gumbel sample.

Proposition 5 (Losslessness). Suppose tree topology and token labels depend only on the committed prefix at step start and are independent of that step's Gumbel noise. Let $y_1, y_2, \ldots$ be committed tokens from prefix $x_{1:t_0}$. Then, for every $i \geq 1$:

$$\Pr\left[y_i = v \mid y_{1:i-1}\right] = q\left(v \mid x_{1:t_0}, y_{1:i-1}\right). \tag{14}$$

Thus the complete output has the same distribution as token-by-token sampling from the target model, for any prefix-measurable tree topology, any draft tokens, and every $T > 0$ (Appendix A.10). Drafter accuracy and topology caching thus govern only acceptance throughput, never generation fidelity.

4 Experiments

4.1 Experimental and Training Setup

We evaluate Qwen3-4B and Qwen3-8B (Yang et al., 2025) against officially released checkpoints of DFlash (Chen et al., 2026) and EAGLE-3 (Li et al., 2026b) in the single-request regime (batch size $= 1$) across eight NVIDIA H800 GPUs ($T = 1$). We report mean acceptance length A (including bonus tokens) and end-to-end wall-clock speedup relative to standard target model autoregressive decoding. Because A aggregates over all decoding rounds across entire benchmarks, per-run variance is negligible, so we report point estimates.

Our default drafter stacks 5 Transformer layers initialized from the target model's bottom 5 layers. Training uses prompts from Open-PerfectBlend (Xu et al., 2024) (1.3M samples) with target-regenerated responses in non-thinking mode. Drafters are trained for 10 epochs using AdamW (lr $= 2 \times 10^{-4}$, global batch size 512) with decay $\gamma = 1$, bag size $K = 6$, candidate pool $n = 12$, beam budget $B = 80$, prefix paths $P = 16$, and 512 anchors per sequence. Repeated-token leaves are enabled on 4B and disabled on 8B due to hardware scaling (§5.4). Detailed training hyperparameters are provided in Appendix A.3.

4.2 Evaluation Benchmarks

The main evaluation suite spans seven benchmarks across three distinct domains: (1) *Mathematical reasoning*: AIME-2025; (2) *Code generation*: HumanEval (Chen et al., 2021), MBPP (Austin et al.,

Method	Math	Code Generation			Dialogue & Instruction			Mean
	AIME'25	HumanEval	MBPP	LCB	MT-Bench	Alpaca	Arena-H	
(a) Acceptance Length (A)								
Target Model: Qwen3-4B								
EAGLE-3	3.793	4.147	3.733	3.742	2.464	2.264	2.527	3.239
DFlash	3.958	4.750	4.405	4.201	3.030	2.935	2.851	3.733
BagDraft	**4.361**	**4.753**	**4.648**	**4.311**	**3.462**	**3.380**	**2.915**	**3.976**
Target Model: Qwen3-8B								
EAGLE-3	3.948	4.303	3.944	4.130	2.705	2.522	2.557	3.444
DFlash	4.016	**4.661**	4.363	4.370	3.058	2.972	2.797	3.748
BagDraft	**4.160**	4.562	**4.455**	**4.614**	**3.344**	**3.241**	**2.834**	**3.887**
(b) Speedup ($\times$)								
Target Model: Qwen3-4B								
EAGLE-3	2.277	2.502	2.263	2.443	1.532	1.327	1.430	1.968
DFlash	2.981	3.574	3.374	3.417	2.370	2.085	1.976	2.825
BagDraft	**3.364**	**3.698**	**3.711**	**3.629**	**2.865**	**2.715**	**2.196**	**3.168**
Target Model: Qwen3-8B								
EAGLE-3	2.322	2.498	2.322	2.576	1.629	1.440	1.406	2.028
DFlash	**3.052**	**3.476**	3.349	3.553	2.367	2.125	1.961	2.840
BagDraft	3.031	3.454	**3.476**	**3.699**	**2.663**	**2.551**	**2.036**	**2.987**

Table 1: **Main evaluation results on Qwen3-4B and Qwen3-8B under stochastic sampling ($T = 1$).** We compare BAGDRAFT against official checkpoints of state-of-the-art drafters across seven benchmarks. (a) **Acceptance length** (A) is the mean number of committed tokens per speculative step; (b) **Speedup** ($\times$) is measured against standard autoregressive decoding at batch size 1 (Equation 1). **Bold** marks the best result per column within each scale.

2021), and LiveCodeBench (Jain et al., 2025); and (3) *Instruction following and dialogue*: MT-Bench (Zheng et al., 2023), AlpacaEval (Dubois et al., 2023), and Arena-Hard-v2 (Li et al., 2024a). Full benchmark descriptions are detailed in Appendix A.4.

5 Results and Analysis

5.1 Training Dynamics

Across target scales and bag sizes ($K \in \{4, 6, 8\}$), top-16 partial coverage rises steadily to stable plateaus, confirming robust convergence without scale-specific retuning; it decreases monotonically as K widens the offset window over a fixed candidate pool (§5.4), with the 8B drafter benefiting most at smaller K. Per-configuration plateau values, full trajectories, bag-entropy contraction, and full-coverage scaling across $n \in \{4, 8, 16, 32\}$ are detailed in Appendix A.5 (Figure 3).

5.2 Main Results

Overall performance. On Qwen3-4B, BAGDRAFT sweeps all seven benchmarks, reaching $A = 3.976$ and $3.168\times$ speedup—outperforming the leading block-diffusion baseline DFlash by $+6.5\%$ in A and $+12.1\%$ in speedup, and autoregressive EAGLE-3 by $+22.8\%$ and $+61.0\%$. On Qwen3-8B, it maintains the lead with $A = 3.887$ and $2.987\times$ ($+3.7\%$/$+5.2\%$ over DFlash), topping 6 of 7 benchmarks in A and 5 of 7 in speedup. The speedup margin narrows from $+12.1\%$ (4B) to $+5.2\%$ (8B) not from algorithmic decline but from hardware economics: per-node verification cost roughly doubles from 4B to 8B (§5.4), favoring sparser trees on larger models.

Domain-specific performance. Gains are most pronounced in open-ended, high-entropy dialogue tasks (MT-Bench, AlpacaEval, Arena-Hard-v2), exceeding DFlash speedups by $+20.9\%$ on MT-

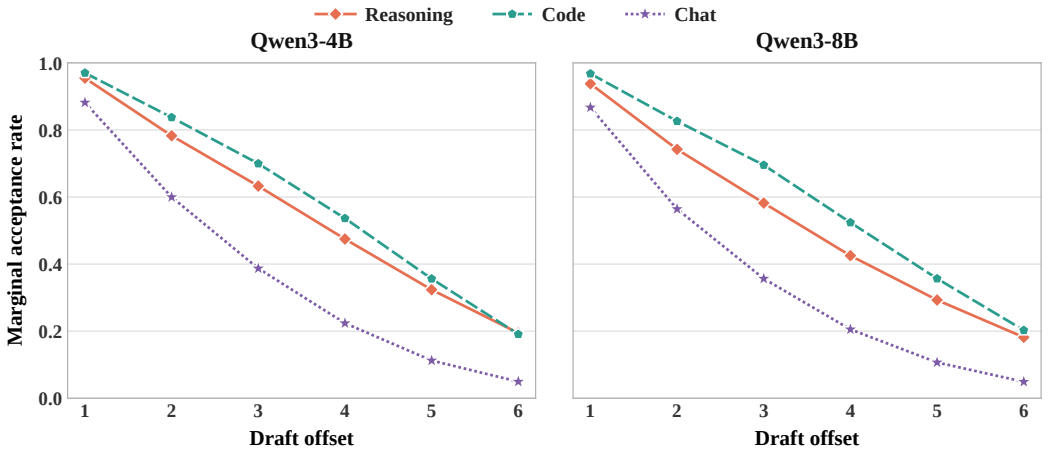


Figure 2: **Marginal acceptance rate across draft offsets** for BAGDRAFT on Qwen3-4B (left) and Qwen3-8B (right) across three task domains (Reasoning, Code, and Chat). Acceptance decays monotonically with offset across all domains and scales, with the high-entropy Chat domain decaying fastest.

Bench ($2.865\times$) and $+30.2\%$ on AlpacaEval ($2.715\times$). Here, rigid slot predictions in block diffusion suffer from spatial misalignments, whereas our shared distribution flexibly pools capacity to identify upcoming tokens without premature position commitments. In structured code and reasoning tasks, BAGDRAFT consistently matches or exceeds DFlash, achieving $3.629\times$/$3.699\times$ speedup on LiveCodeBench (4B/8B) and $A = 4.361$ on AIME-2025 ($+10.2\%$).

5.3 PER-DEPTH ACCEPTANCE ANALYSIS

To dissect draft capacity across positions, we measure per-depth marginal acceptance averaged across our seven benchmarks on Qwen3-4B. BAGDRAFT leads DFlash at offsets 0–3 and trails only at offsets 4–5, precisely mirroring the $\gamma = 1$ loss allocation (86.7% of loss mass on offsets 0–1, 1.6% on offsets 4–5); full per-offset rates are tabulated in Appendix A.7. Figure 2 shows this decay is strongly domain-dependent—collapsing to $\sim$0.05 on high-entropy Chat while sustaining $\sim$0.19 on Code and Reasoning (explaining the speedup gradient in §5.2)—with 4B and 8B curves remaining virtually indistinguishable.

5.4 COST BREAKDOWN AND ABLATIONS

Marginal verification latency scales from 0.0034 to 0.0074 ms/node (4B$\rightarrow$8B), placing the "free-tree" budget ceiling near 300 nodes; beyond it (e.g., $P = 32$ inflates the tree to 452.7 nodes) speedup degrades despite minor A gains, so we adopt $P = 16$. Ablations confirm the design: bag capacity $K = 6$ is the effective ceiling ($K = 8$ degrades A by $\sim$10% under fixed γ), the prefix-path budget is monotone in A but throughput-capped at $P = 16$, and repeated-token leaves stay throughput-positive only on 4B. The stage-level latency breakdown is given in Appendix A.8.

6 CONCLUSION

We presented **BagDraft**, which recasts speculative drafting as a next-bag-of-tokens task: a single shared logit vector under weighted multi-hot cross-entropy emits unordered candidates, while prefix-monotone Plackett–Luce pruning and exact target verification reconstruct order losslessly at inference—so discarding sequence order requires no parametric recovery training. Across Qwen3-4B and Qwen3-8B, this yields consistent gains in both acceptance length and wall-clock speedup (up to $3.17\times$ on 4B and $2.99\times$ on 8B) over official DFlash and EAGLE-3 baselines. BAGDRAFT thus shows that state-of-the-art acceleration needs no dedicated per-position heads or autoregressive recurrence; limitations and future work are discussed in Appendix A.1.

REFERENCES

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.

Marianne Arriola, Aaron Gokaslan, Justin Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, volume 2025, pp. 50726–50753, 2025.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

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

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

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

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

Zhuoming Chen, Avner May, Ruslan Svirschevski, Yuhsun Huang, Max Ryabinin, Zhihao Jia, and Beidi Chen. Sequoia: Scalable and robust speculative decoding. *Advances in Neural Information Processing Systems*, 37:129531–129563, 2024.

Tianqi Du, Lizhe Fang, Weijie Yang, Chenheng Zhang, Zeming Wei, Yifei Wang, and Yisen Wang. Autoregressive models rival diffusion models at any-order generation. *arXiv preprint arXiv:2601.13228*, 2026.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

Itai Gat, Heli Ben-Hamu, Marton Havasi, Daniel Haziza, Jeremy Reizenstein, Gabriel Synnaeve, David Lopez-Paz, Brian Karrer, and Yaron Lipman. Set block decoding is a language model inference accelerator. *arXiv preprint arXiv:2509.04185*, 2025.

Anastasios Gerontopoulos, Spyridon Gidaris, and Nikos Komodakis. Multi-token prediction needs registers. *Advances in Neural Information Processing Systems*, 38:60189–60214, 2026.

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

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.

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Ashish Khisti, MohammadReza Ebrahimi, Hassan Dbouk, Arash Behboodi, Roland Memisevic, and Christos Louizos. Multi-draft speculative sampling: Canonical decomposition and theoretical limits. In *International Conference on Learning Representations*, volume 2025, pp. 1593–1633, 2025.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*, 2025.

Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International conference on machine learning*, pp. 3499–3508. PMLR, 2019.

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

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*, 2024a.

Tianyi Li, Yaxin Luo, Xinyi Shang, and Zhiqiang Shen. Dartree: Speculative diffusion decoding with autoregressive draft trees. *arXiv preprint arXiv:2608.13524*, 2026a.

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

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *Advances in Neural Information Processing Systems*, 38:136737–136756, 2026b.

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

Divyat Mahajan, Sachin Goyal, Badr Youbi Idrissi, Mohammad Pezeshki, Ioannis Mitliagkas, David Lopez-Paz, and Kartik Ahuja. Beyond multi-token prediction: Pretraining llms with future summaries. In *International Conference on Learning Representations*, volume 2026, pp. 64342–64364, 2026.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, et al. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pp. 932–949, 2024.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *Advances in Neural Information Processing Systems*, 38:50608–50646, 2026.

Bowen Peng, Théo Gigant, and Jeffrey Quesnelle. Efficient pre-training with token superposition. *arXiv preprint arXiv:2605.06546*, 2026.

Liran Ringel and Yaniv Romano. Accelerating speculative decoding with block diffusion draft trees. *arXiv preprint arXiv:2604.12989*, 2026.

Subham Sekhar Sahoo, Lingjie Chen, Khiem Pham, Jonathan Geuter, Chaitanya Dwivedi, Varad Pimpalkhute, Yash Akhauri, Alexander Moreno, Mikhail Yurochkin, Zhenting Wang, et al. Unlocking lossless speedups in llms via discrete diffusion. *arXiv preprint arXiv:2609.04010*, 2026.

Luohe Shi, Zuchao Li, Lefei Zhang, Baoyuan Qi, Guoming Liu, and Hai Zhao. Scaling llm speculative decoding: Non-autoregressive forecasting in large-batch scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32947–32955, 2026.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. Spectr: Fast speculative decoding via optimal transport. *Advances in Neural Information Processing Systems*, 36:30222–30242, 2023.

Ziteng Sun, Uri Mendlovic, Yaniv Leviathan, Asaf Aharoni, Jae H Ro, Ahmad Beirami, and Ananda Theertha Suresh. Block verification accelerates speculative decoding. In *International Conference on Learning Representations*, volume 2025, pp. 24965–24994, 2025.

Jikai Wang, Yi Su, Juntao Li, Qingrong Xia, Zi Ye, Xinyu Duan, Zhefeng Wang, and Min Zhang. Opt-tree: Speculative decoding with adaptive draft tree structure. *Transactions of the Association for Computational Linguistics*, 13:188–199, 2025.

Yepeng Weng, Qiao Hu, Xujie Chen, Li Liu, Dianwen Mei, Huishi Qiu, Jiang Tian, et al. Traversal verification for speculative tree decoding. *Advances in Neural Information Processing Systems*, 38:77899–77924, 2026.

Ephrem Wu. Ceiling-clipped acceptance histograms indicate stranded speed-up in block-diffusion speculative decoding. *arXiv preprint arXiv:2608.30427*, 2026.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 7655–7671, 2024.

Tengyu Xu, Eryk Helenowski, Karthik Abinav Sankararaman, Di Jin, Kaiyan Peng, Eric Han, Shaoliang Nie, Chen Zhu, Hejia Zhang, Wenxuan Zhou, et al. The perfect blend: Redefining rlhf with mixture of judges. *arXiv preprint arXiv:2409.20370*, 2024.

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

Jiebin Zhang, Zhenghan Yu, Song Liu, Eugene J Yu, Zheng Li, Dawei Zhu, Jiangshan Duo, Weimin Xiong, Yifan Song, Guanghua Yu, et al. Dflare: Scaling up draft capacity for block diffusion speculative decoding. *arXiv preprint arXiv:2606.02091*, 2026a.

Lefan Zhang, Xiaodan Wang, Yanhua Huang, and Ruiwen Xu. Learning harmonized representations for speculative sampling. In *International Conference on Learning Representations*, volume 2025, pp. 35367–35388, 2025.

Shuai Zhang, Huachuan Qiu, Hongliang He, and Yong Dai. Cost-aware diffusion draft trees for speculative decoding. *arXiv preprint arXiv:2606.01813*, 2026b.

Yaojie Zhang, Linfeng Zhang, Bin Cui, and Xupeng Miao. Dflow: Enabling verifier information flow in block diffusion speculative decoding, 2026c. 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 Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Zayd MK Zuhri, Erland Hilman Fuadi, and Alham Fikri Aji. Predicting the order of upcoming tokens improves language modeling. *arXiv preprint arXiv:2508.19228*, 2025.

A Appendix

A.1 Limitations and Future Work

While BagDraft achieves substantial acceleration, its primary limitation stems from the without-replacement formulation (Assumption 1), which assigns zero probability to windows containing duplicate tokens. Consequently, token repetitions are handled via heuristic, unranked leaves that cannot receive descendants (§3.4), capping the accepted prefix whenever the true continuation repeats—a

frequent pattern in structured text and code (e.g., brackets, indentation). Furthermore, because per-node verification latency scales with model size (0.0074 ms/node on 8B vs. 0.0034 ms/node on 4B), these auxiliary leaves are selectively disabled on Qwen3-8B to remain within the free-tree budget ceiling. Future work includes developing unified ranking distributions that natively support multi-sets—such as Plackett–Luce variants with geometric re-insertion—to incorporate repeated tokens directly into the prefix-monotone beam search without ad-hoc width compensation.

A.2 AI USE STATEMENT

During the preparation of this manuscript, we used a large language model (LLM) to assist with language polishing, including improving wording, grammar, and readability. All research ideas, methodological design, experimental implementation, and analysis of results were carried out independently by the authors, who take full responsibility for the entire content of the paper.

A.3 TRAINING CONFIGURATION

The drafter backbone consists of the bottom 5 Transformer layers of the target model, taking as input the target's hidden states captured at 5 evenly spaced layers and the bag size $K = 6$ future-slot supervision signal under the offset-weighted multi-hot cross-entropy loss with decay constant $\gamma = 1.0$ (Section 3). Each training sequence samples 512 anchor positions (`num_anchors=512`) under the discrete anchor-block masking scheme. Training uses the AdamW optimizer with learning rate 2×10^{-4}, a 4% linear warmup ratio, zero weight decay, gradient clipping at $\|\nabla\| \leq 1.0$, and bf16 mixed precision, run for 10 epochs with a global batch size of 512 (local batch size 8 per GPU, full-shard FSDP across 8 NVIDIA H800 GPUs). The same recipe is applied unchanged across bag sizes $K \in \{4, 6, 8\}$ and both target scales (Qwen3-4B and Qwen3-8B), with only the bag size and, for a subset of scaling runs, the accelerator topology and global batch size varying.

A.4 EVALUATION BENCHMARK AND DATASET DETAILS

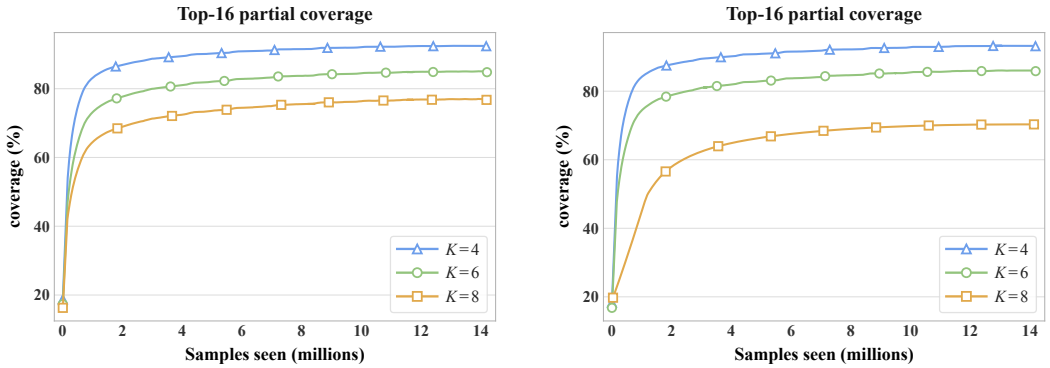
Training Dataset Composition. For training, we use Open-PerfectBlend (Xu et al., 2024), an open-sourced dataset derived from PerfectBlend containing 1.3 million samples across four functional categories: general multi-turn conversation (17.6%), mathematical reasoning (39.4%), code synthesis (38.9%), and general instruction-following (4.1%). Prompts are filtered to remove degenerate samples, and completions are regenerated by each target model in non-thinking mode using recommended sampling parameters.

Detailed Descriptions of Evaluation Benchmarks. The evaluation suite spans seven representative benchmarks across three core domains:

- **AIME-2025**: A competition-level mathematics benchmark consisting of 30 challenging problems from the 2025 American Invitational Mathematics Examination, designed to test multi-step symbolic derivation and numerical calculation.
- **HumanEval** (Chen et al., 2021) & **MBPP** (Austin et al., 2021): Standard function-level Python programming benchmarks containing 164 and 256 programming tasks, evaluated against functional unit tests.
- **LiveCodeBench** (Jain et al., 2025): A contamination-resistant coding benchmark continuously collected from competitive programming contests (LeetCode, AtCoder, Codeforces), reflecting realistic coding difficulties.
- **MT-Bench** (Zheng et al., 2023): A multi-turn dialogue benchmark encompassing 8 common categories, evaluated via pairwise LLM-as-a-judge scoring.
- **AlpacaEval** (Dubois et al., 2023) & **Arena-Hard-v2** (Li et al., 2024a): Rigorous instruction-following benchmarks evaluating response quality against strong baseline outputs using automated GPT-4-Turbo judges.

A.5 ADDITIONAL TRAINING DIAGNOSTICS

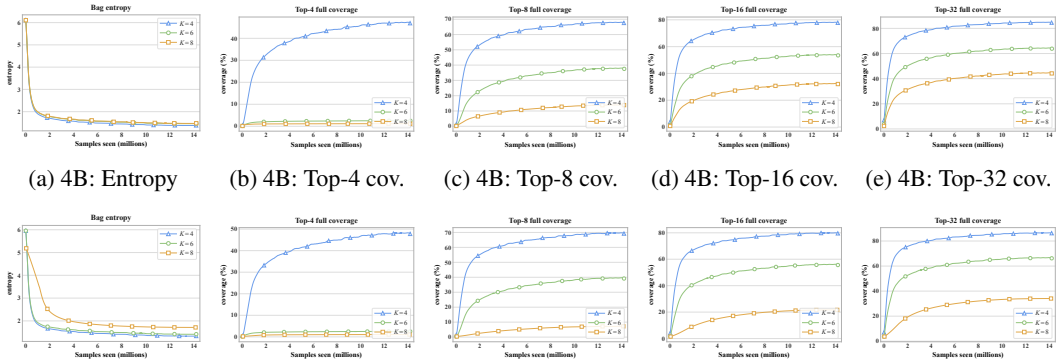
Figure 4 complements Figure 3 with two further training-time diagnostics: bag entropy, and full-coverage rate (the fraction of anchors whose entire valid offset window is covered by the top-n bag



(a) Qwen3-4B: Top-16 partial coverage

(b) Qwen3-8B: Top-16 partial coverage

Figure 3: **Training dynamics of BagDraft across target model scales and bag sizes.** Curves track top-16 partial coverage against samples seen (millions), with colors distinguishing bag size $K \in \{4, 6, 8\}$. Points represent logging-window summaries; curves are smoothed solely for visualization. Coverage measures the proportion of valid offset targets covered by the top-16 bag candidates, a training-time analogue of inference-time acceptance.



(a) 4B: Entropy (b) 4B: Top-4 cov. (c) 4B: Top-8 cov. (d) 4B: Top-16 cov. (e) 4B: Top-32 cov.

(f) 8B: Entropy (g) 8B: Top-4 cov. (h) 8B: Top-8 cov. (i) 8B: Top-16 cov. (j) 8B: Top-32 cov.

Figure 4: **Additional training diagnostics for BagDraft across model scales.** Bag entropy and full-coverage rate across candidate-pool sizes $n \in \{4, 8, 16, 32\}$ for Qwen3-4B (top row) and Qwen3-8B (bottom row). Curves are plotted against samples seen (millions), with colors distinguishing bag sizes $K \in \{4, 6, 8\}$, complementing the primary convergence metrics in Figure 3.

candidates) swept over $n \in \{4, 8, 16, 32\}$. Top-16 partial coverage converges to stable plateaus of 92.5%, 85.0%, and 77.0% on 4B and 93.3%, 86.1%, and 72.1% on 8B for $K \in \{4, 6, 8\}$ respectively. The decreasing bag entropy confirms that optimization concentrates probability mass onto a compact set of high-confidence tokens. Full-coverage rate is naturally lower than the top-16 partial coverage reported in the main text, since it requires the entire offset window rather than a single offset to be covered, and it decreases as K grows for the same reason discussed in Section 4.1.

A.6 Draft Backbone Depth

We further ablate the depth of the draft backbone—the number of bottom Transformer layers of the target model reused by the drafter—at $\{1, 2, 5\}$ layers, holding all other hyperparameters fixed. Figure 5 reports the converged top-16 partial coverage for both target scales across bag sizes $K \in \{4, 6, 8\}$. Coverage increases monotonically with depth at every (scale, K): moving from a 1-layer to a 5-layer backbone lifts coverage by roughly 12–15 points, and the gap widens as K grows, since a deeper backbone is needed to span the wider offset window. This motivates our default 5-layer drafter; shallower backbones trade accuracy for a smaller drafting cost and remain a viable operating point under tight latency budgets.

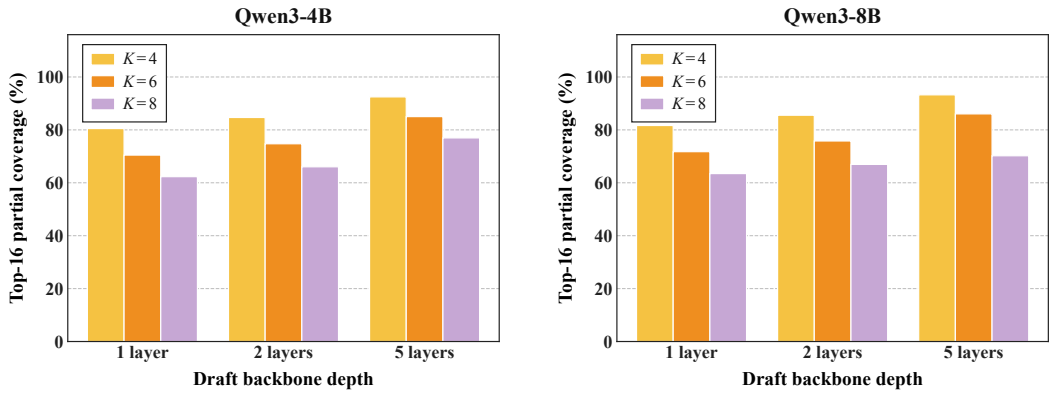


Figure 5: **Effect of draft backbone depth on convergence.** Converged top-16 partial coverage (mean over the final logged window) as a function of the number of reused Transformer layers (1/2/5), grouped by bag size K, for Qwen3-4B and Qwen3-8B. Deeper backbones yield uniformly higher coverage, with the margin growing at larger K.

A.7 Per-Depth Acceptance Details

This appendix expands the per-depth analysis of §5.3. Unless noted, acceptance rates are marginal per-offset rates $r_k = \Pr[\text{offset } k \text{ lies on the committed chain} \mid \text{offset } k \text{ proposed}]$.

Table 2 reports per-offset acceptance averaged over the seven main benchmarks on Qwen3-4B. BagDraft leads DFlash by $+0.13$ at offsets 0–2, the lead narrows to $+0.07$ at offset 3, and the sign flips at offset 4—exactly where the $\gamma = 1$ weight sequence $\tilde{w}_k \propto e^{-k}$ drops the allocated loss mass below 1% (offset 4: 1.2%, offset 5: 0.4%). The steeper BagDraft profile (decay rate ≈ 0.83/offset vs. DFlash's flatter ≈ 0.86) is the direct fingerprint of exponential offset weighting: early offsets, which gate prefix verification, receive the bulk of representational capacity, while deep offsets are intentionally undertrained. The summed profile $1 + \sum_k r_k = 3.985$ recovers the mean acceptance length $A = 3.976$ from Table 1 up to bonus-token rounding, confirming the diagnostic is drawn from the same regime as the headline evaluation.

Offset k	0	1	2	3	4	5	6
$\gamma = 1$ loss mass	63.4%	23.3%	8.6%	3.2%	1.2%	0.4%	—
BagDraft r_k	**0.930**	**0.728**	**0.556**	**0.394**	0.247	0.131	—
DFlash r_k	0.801	0.591	0.433	0.327	**0.253**	**0.196**	0.148
Difference	+0.129	+0.137	+0.123	+0.066	−0.005	−0.065	—

Table 2: Per-offset marginal acceptance rates averaged over the seven main benchmarks (Qwen3-4B). **Bold** marks the higher value at each offset; the crossover at offset 4 coincides with the offset where $\gamma = 1$ loss mass falls below 1%.

Figure 2 further decomposes this profile by task domain for both scales: Chat collapses to ~ 0.05 by offset 5 while Code and Reasoning sustain ~ 0.19, and the 4B and 8B curves are nearly identical, indicating the drafter's per-offset capacity allocation transfers across target scales without retuning.

A.8 Cost Breakdown

This appendix expands the cost discussion of §5.4; latency figures are mean per-step wall-clock decompositions collected under the same single-request profiling harness as the main results.

In this length regime the target forward is launch-bound and stays essentially invariant to tree size, so the marginal per-node cost is set by the draft forward and the tree-select/build/verify group. This per-node cost roughly doubles from 4B to 8B, which lowers the "free-tree" budget ceiling to ∼300 nodes on 8B and drives the width-compensation budget: because the non-target host-side work is

a small fraction of each step, the residual throughput headroom is only a few percent and the 8B bottleneck is the drafter itself, not the verification implementation.

A.9 Proofs for the Shared-Logit Objective

Proof of Proposition 1. For $q, p \in \Delta^{V-1}$, let $H(q) = -\sum_v q(v) \log q(v)$ and $\mathrm{KL}(q\|p) = \sum_v q(v) \log(q(v)/p(v))$, with $0 \log 0 = 0$. Then $\mathbb{E}_{y\sim q}[-\log p(y)] = H(q) + \mathrm{KL}(q\|p)$. Since the first term is independent of p, Gibbs' inequality gives the unique minimizer $p = q$.

For independent heads, the risk is $\sum_k w_k \big(H(q_k) + \mathrm{KL}(q_k\|p_k)\big)$. Every term contains only p_k, and $w_k > 0$, so its unique minimizer is $p_k^\star = q_k$. For the shared distribution, the risk is

$$\sum_k \tilde{w}_k \mathbb{E}_{y_k\sim q_k}[-\log s(y_k)] = -\sum_{v=1}^{V} \left(\sum_k \tilde{w}_k q_k(v)\right) \log s(v) \tag{15}$$

$$= H(\bar{q}) + \mathrm{KL}(\bar{q}\|s), \tag{16}$$

where $\bar{q} = \sum_k \tilde{w}_k q_k$. Gibbs' inequality yields $s^\star = \bar{q}$. If a minimizer lies on the boundary of the simplex, a sequence of finite logits approaches it; this does not affect the comparison between the population optima.

Proof of Corollary 1. With uniform weights, $\Phi(\mathcal{Q}) = K^{-1} \sum_k q_k$, which is invariant to a permutation of the summands. For positive weights, let $U = \{u \in \mathbb{R}^V : \sum_v u(v) = 0\}$, so $\dim U = V - 1$. Given $u_1, \ldots, u_{K-1} \in U$, define

$$q_0' = q_0 + \frac{1}{\tilde{w}_0} \sum_{k=1}^{K-1} \tilde{w}_k u_k, \qquad q_k' = q_k - u_k \quad (k \geq 1). \tag{17}$$

For sufficiently small perturbations, full support ensures $\mathcal{Q}' \in (\Delta^{V-1})^K$, and equation 17 preserves Φ. These perturbations form a convex subset of the fiber with relative dimension $(K-1)(V-1)$. If an inverse Ψ existed on a full-support open set, two distinct sufficiently close points $\mathcal{Q}$ and $\mathcal{Q}'$ in that set with $\Phi(\mathcal{Q}) = \Phi(\mathcal{Q}')$ would imply $\mathcal{Q} = \mathcal{Q}'$, a contradiction. Finally, for distinct deterministic labels, equation 7 gives $s^\star(y_k) = \tilde{w}_k$. The weights $\exp(-k/\gamma)$ strictly decrease with k when $0 < \gamma < \infty$, proving the ranking claim.

A.10 Proofs for Permutation-Tree Construction and Verification

Proof of Proposition 2 (Path probability). Let C be a candidate pool with $s_i > 0$ for every $i \in C$ and $S_C = \sum_{i\in C} s_i$. Under sequential sampling without replacement, the first draw has probability s_{i_1}/S_C. After drawing $i_1, \ldots, i_{r-1}$, the remaining pool mass is $S_C - \sum_{u<r} s_{i_u}$, which is strictly positive for $r \leq |C|$. Hence

$$\Pr[i_r \mid i_1, \ldots, i_{r-1}] = \frac{s_{i_r}}{S_C - \sum_{u<r} s_{i_u}}. \tag{18}$$

Applying the chain rule to a sequence $\pi = (i_1, \ldots, i_d)$ of distinct elements gives

$$\hat{p}(\pi) = \prod_{r=1}^{d} \frac{s_{i_r}}{S_C - \sum_{u<r} s_{i_u}}. \tag{19}$$

Because equation 10 is a product of normalized conditional probabilities, it sums to one over all ordered length-d sequences. For $d = 1$, its marginal is s/S_C on C. This is the Plackett–Luce ranking distribution induced by the pool-normalized weights. Equivalently, independently perturbing each $\log s_i$ by a standard Gumbel variable and sorting the perturbed scores in descending order produces the same distribution. Within a fixed parent, all children share the denominator in equation 10; their relative order is therefore the order of their s_i values.

Proof of Proposition 3 (Prefix monotonicity). Let $\pi = (i_1, \ldots, i_d)$ with $d < |C|$, and let $\pi' = (\pi, i_{d+1})$ for an unused $i_{d+1} \in C$. Equation 10 yields

$$\frac{\hat{p}(\pi')}{\hat{p}(\pi)} = \frac{s_{i_{d+1}}}{S_C - \sum_{u\leq d} s_{i_u}} = \frac{s_{i_{d+1}}}{\sum_{i\in C\setminus\{i_1,\ldots,i_d\}} s_i}. \tag{20}$$

The numerator is one positive summand of the denominator. The ratio thus lies in $(0, 1]$, and it equals one only when i_{d+1} is the sole remaining element, namely when $d + 1 = |C|$. Hence every path extension has no larger probability than its prefix. When the tree depth $K < |C|$, the inequality is strict on every tree edge.

Proof of Proposition 4 (Prefix-local monotonicity in P). Fix a decoding-step prefix, and hence fix the shared distribution s, candidate-pool size, beam width, maximum depth, and width offset. Suppose the width-compensation leaves have no global cap and repeated-token leaves are disabled. Let $\mathcal{T}(P)$ be the tree constructed from the top P beam paths.

First, the parent set is monotone in P. Beam-search paths, the pinned depth-one nodes, and the reach-probability ranking are all determined before selecting P. Increasing P adds paths to the selected prefix of this fixed ranking, so the union of their deduplicated prefixes can only grow.

Second, the appended-leaf set is monotone. The children appended to a selected parent depend only on that parent, its ancestor chain, its depth, and its children already present in the beam tree. None of these quantities depends on P or on the selection of other parents. Since appended leaves do not themselves acquire children and there is no global cap, a parent common to $\mathcal{T}(P)$ and $\mathcal{T}(P + 1)$ receives exactly the same appended leaves; newly selected parents add leaves without removing any existing node. Therefore,

$$\mathcal{T}(P) \subseteq \mathcal{T}(P + 1). \tag{21}$$

Couple target samples on the two trees by reusing the Gumbel noise for every common tree node, identified by its token-prefix path, and drawing independent noise only for nodes unique to $\mathcal{T}(P+1)$. The target distribution at a common node is identical in both trees because the tree-attention mask permits that node to attend only to the committed prefix and its own ancestor chain. This coupling preserves both marginal distributions. Let ρ be the longest accepted chain in $\mathcal{T}(P)$. Every node of ρ also occurs in $\mathcal{T}(P + 1)$, and every acceptance event on ρ depends on the unchanged parent sample and edge token. Thus ρ remains accepted in $\mathcal{T}(P + 1)$. Because sibling tokens are distinct, newly added children cannot compete with an already accepted edge; they can only extend a chain that previously terminated. Consequently,

$$A(\mathcal{T}(P)) \leq A(\mathcal{T}(P + 1)) \tag{22}$$

pointwise under the coupling, and taking expectations proves the claim.

This is a statement for a fixed step-start prefix, not for a trajectory average. A larger P can advance the same lossless output stream by more tokens per step and therefore changes the distribution of prefixes at later step boundaries. The prefix-local result neither orders these later prefixes nor guarantees a monotone empirical trajectory average or speedup.

Proof of Proposition 5 (Losslessness). Fix a committed-token index i and a realized prior committed sequence $y_{1:i-1}$, and write $h = (x_{1:t_0}, y_{1:i-1})$. We show that

$$\Pr[y_i = v \mid y_{1:i-1}] = q(v \mid h). \tag{23}$$

There is a unique tree position J_i that produces y_i. Within a speculative step, the root sample produces the first committed token. A child is accepted exactly when its distinct token equals its parent sample, so at most one child is accepted at each parent; the accepted chain continues until no child matches or the maximum depth is reached, and the terminal-position sample is committed as the bonus token. Given $y_{1:i-1}$, this recursion determines all earlier step boundaries and accepted-chain positions, and hence determines J_i.

The context of J_i is exactly h. If J_i lies on an accepted chain, every token on its ancestor chain equals the corresponding token already committed in that step, while the key/value cache supplies the step-start committed prefix. If J_i is the root of a later step, that step starts directly from h. The tree-attention mask excludes all side branches, so the target distribution at J_i is $q(\cdot \mid h)$.

Finally, the target token at J_i is an exact Gumbel-max draw from that distribution. The positions that produced $y_1, \ldots, y_{i-1}$ are distinct from J_i, and the event $\{y_{1:i-1}\}$ depends only on their Gumbel noise. By the assumed independence of Gumbel noise across tree positions and speculative steps, conditioning on this event leaves the noise at J_i independent and standard Gumbel. Therefore its

sample has distribution $q(\cdot \mid h)$, which proves equation 23. Applying the chain rule over i gives the same joint distribution as token-by-token sampling from the target model.

Temperature only changes each local distribution to $q_j = \mathrm{softmax}(z_j^{\mathrm{target}}/T)$, so the argument holds for every $T > 0$. The tree controls which precomputed positions are available for a match, but not the conditional distribution of any committed token; any tree topology and draft tokens that are measurable with respect to the step-start prefix and independent of that step's Gumbel noise therefore preserve losslessness.