Teaching Diffusion to Speculate Left-to-Right

Lexington Whalen*, **Yuki Ito, Ryo Sakamoto**

SB Intuitions, Tokyo, Japan
lexington.whalen@sbintuitions.co.jp, yuki.ito@sbintuitions.co.jp, ryo.sakamoto@sbintuitions.co.jp

Abstract

Large language models (LLMs) achieve remarkable performance across a wide range of tasks, but their autoregressive decoding process incurs substantial inference costs due to inherently sequential token generation. Speculative decoding addresses this bottleneck by employing a lightweight draft model to propose multiple future tokens that are subsequently verified in parallel by a larger target model. Recent work has demonstrated that diffusion language models are well suited for this setting, as they can generate entire blocks of draft tokens in parallel and thereby alleviate the sequential constraints of autoregressive drafting. A subtlety of this regime is that block-diffusion drafters generate tokens *bidirectionally* within a block, whereas verification is performed by an autoregressive target model that evaluates tokens in a strictly *left-to-right manner*, leaving a gap between the symmetric training-time objective and the asymmetric verification-time reward. In this work, we offer an empirical analysis of three training-time interventions that narrow this gap: token positional weighting, a first-error focal loss that targets the position that breaks the accepted prefix within each block, and a chain loss term that substitutes a differentiable surrogate for the expected accepted length. The three interventions act along orthogonal axes (position, block-conditional first error, joint prefix) and compose additively; they are likewise orthogonal to test-time alignment mechanisms such as multi-draft self-selection, with which they can in principle be combined. Across four target models and six reasoning, code, and dialogue benchmarks, the three interventions raise accepted draft length by 21–76% per benchmark over a position-uniform baseline, without adding additional forward passes and without changing the inference pipeline or the rejection-sampling exactness contract.

1 Introduction

Large language models (LLMs) have become a central computational primitive across applications ranging from conversational assistants and code generation (Chen et al. 2021; Rozière et al. 2023) to retrieval-augmented question answering, multi-step agent systems, and long-horizon reasoning (OpenAI 2024; DeepSeek-AI et al. 2025). As these systems mature, the dominant cost in their lifecycle has shifted from training to inference: a model is trained once but served continuously, and per-query cost is amplified by output length, by recursive agentic invocations, and by the long chains of thought emitted by recent reasoning models. Reducing the latency and per-token cost of LLM inference is therefore one of the most consequential systems problems in contemporary machine learning.

Single-stream LLM decoding is bottlenecked by memory bandwidth rather than arithmetic throughput: each autoregressive step streams the entire parameter set from high-bandwidth memory (HBM) to produce a single token, an imbalance that widens with every hardware generation as compute scales faster than HBM (Pope et al. 2023). The standard responses—quantization (Dettmers et al. 2022; Frantar et al. 2023; Lin et al. 2024), sparsity (Frantar and Alistarh 2023), distillation (Hinton, Vinyals, and Dean 2015), and system-level techniques such as FlashAttention (Dao et al. 2022), PagedAttention, and continuous batching (Kwon et al. 2023; Yu et al. 2022)—all reduce the cost of a single forward pass. Speculative decoding (Leviathan, Kalman, and Matias 2023; Chen et al. 2023) instead reduces the *number* of serialized forward passes required: a cheap drafter proposes K candidate tokens, the target verifies them in a single parallel pass, and a rejection-sampling step accepts the longest prefix consistent with the target distribution. The procedure is exact—samples are statistically identical to those of the target model—which eliminates the quality–versus–speed trade-off that complicates the other techniques and makes the central question purely one of drafter design: how to construct a draft distribution that is simultaneously cheap to evaluate and well aligned with the target.

The EAGLE family (Li et al. 2024b,a, 2025b) pursues this goal by drafting in *feature* space rather than token space, coupling a small auxiliary head to the target's own hidden states. Successive iterations have introduced dynamic tree expansion and multi-layer feature aggregation, and EAGLE-3 has become the de facto baseline for production speculative decoding. The EAGLE drafter nevertheless remains *autoregressive* within each speculation step, requiring token predictions to be generated sequentially and thereby limiting its maximum achievable speedup.

In this work we focus on *block-diffusion drafters*—of which DFlash (Chen, Liang, and Liu 2026) is a prominent recent example—a class that replaces autoregressive feature prediction with a *diffusion-style* parallel block decoder. Conditioned on a configurable set of intermediate target hidden states, such a drafter emits an entire K-token block in a single non-autoregressive forward pass; the block is verified

*Corresponding author.

Technical Report

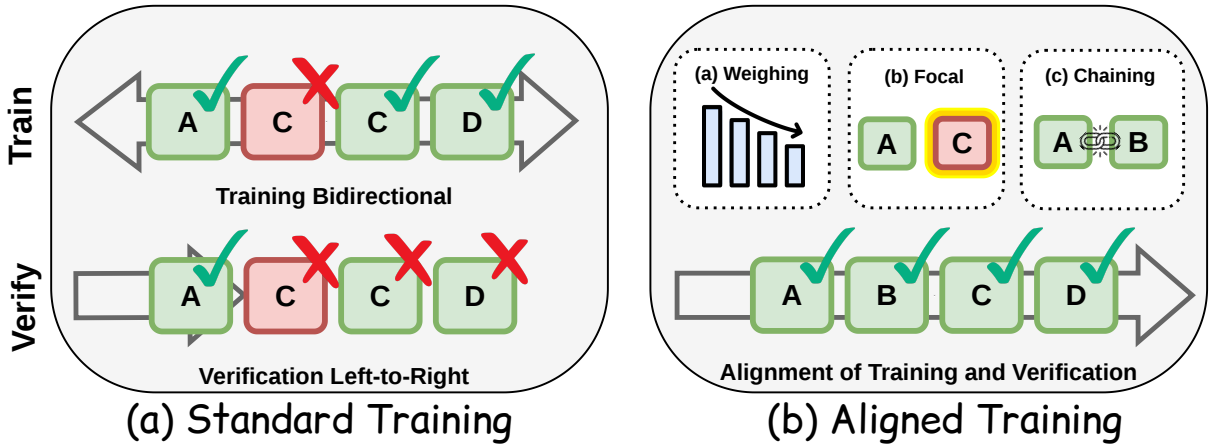

Figure 1: The training–verification mismatch in diffusion-based speculative drafters and the three training-time interventions we analyse. **(a) Standard training.** The drafter is trained with fully bidirectional attention over the K-token block, so every position is conditioned symmetrically on every other. At inference time, however, the target verifies the block strictly left-to-right: a single early rejection (here, position 2) truncates the entire suffix. The drafter's `accd` matches the ground-truth `abcd` at positions 3 and 4, but both are discarded along with the rejected position-2 token. **(b) Aligned training.** We analyze three complementary training-time interventions that bring the drafter's objective into closer correspondence with the causal acceptance contract: position-wise loss decay (*Weighting*), an auxiliary cross-entropy term targeting the first mispredicted position of each block (*Focal*), and a differentiable surrogate for the joint prefix acceptance probability (*Chaining*). Together they raise the expected accepted prefix length and recover the per-token correctness that bidirectional training already provides.

by the target under the standard rejection-sampling contract, preserving exactness. With $K = 16$, this raises the per-step ceiling by more than a factor of three over EAGLE-3 while retaining competitive per-token acceptance through multi-layer conditioning. Pairing a block-diffusion drafter with an autoregressive target also surfaces a representational asymmetry that is less pronounced in the autoregressive-drafter regime: diffusion models are trained to denoise blocks under fully bidirectional attention, so each position conditions on context from both directions, whereas the target verifies strictly left-to-right and accepts only the longest causally consistent prefix. The drafter must therefore allocate its predictive capacity asymmetrically—early positions are disproportionately load-bearing, since a single early divergence truncates all subsequent draft tokens regardless of their quality—an objective in tension with the symmetric denoising loss inherited from the diffusion formulation. The present work studies this tension empirically by analysing three training-time interventions that reshape the position-wise loss profile along orthogonal axes; the analysis is complementary to test-time alignment mechanisms developed for the same drafter class, such as ddTree (Ringel and Romano 2026).

2 Related Work

Efficient LLM inference. A broad portfolio of techniques mitigates the memory-bandwidth cost of single-stream decoding: low-bit quantization (Dettmers et al. 2022; Frantar et al. 2023; Lin et al. 2024; Xiao et al. 2023), sparsity (Frantar and Alistarh 2023; Sun et al. 2024), distillation (Hinton, Vinyals, and Dean 2015; Sanh et al. 2019), FlashAttention (Dao et al. 2022; Dao 2024), and system-level techniques such as PagedAttention, continuous batching, and chunked prefill (Kwon et al. 2023; Yu et al. 2022; Agrawal et al. 2023).

Speculative decoding. Block-parallel decoding was first explored as a deterministic acceleration technique (Stern, Shazeer, and Uszkoreit 2018); the modern, distribution-preserving formulation was introduced concurrently by Leviathan, Kalman, and Matias (2023) and Chen et al. (2023). SpecInfer (Miao et al. 2024) generalised linear-chain drafts to trees verified in parallel under tree-structured attention, substantially raising expected accepted lengths. Subsequent work has expanded the design space along largely orthogonal axes: Medusa (Cai et al. 2024) attaches parallel prediction heads directly to the target, Lookahead Decoding (Fu et al. 2024) sidesteps drafter training via Jacobi-style fixed-point iteration on n-gram trajectories, self-speculative decoding (Zhang et al. 2024a) reuses a subset of target layers as the drafter, and online speculative decoding (Liu et al. 2024b) adapts the drafter continuously to the deployed workload—each accepting the same exactness contract under a different cheap-candidate mechanism.

Feature-level drafters. A particularly successful line draws candidates from the target's own intermediate representations rather than from an independently trained small LM. EAGLE (Li et al. 2024b) introduced feature-level autoregression; EAGLE-2 (Li et al. 2024a) added context-dependent dynamic tree expansion; and EAGLE-3 (Li et al. 2025b) augmented the drafter with multi-layer feature aggregation, establishing the de facto baseline for production speculative decoding. Related designs include Hydra (Ankner et al. 2024), GliDe with CaPE (Du et al. 2024), Kangaroo (Liu et al. 2024a), and HASS (Zhang et al. 2024b). All remain

autoregressive within a speculation step, so their empirical horizons are bounded by tree depth rather than by the cost of a single drafter pass.

Diffusion and non-autoregressive language modelling. Parallel-block generation has a long history outside speculative decoding: non-autoregressive translation (Gu et al. 2018) first demonstrated single-pass sequence generation, and iterative refinement (Lee, Mansimov, and Cho 2018) together with masked-LM decoders such as Mask-Predict (Ghazvininejad et al. 2019) and SUNDAE (Savinov et al. 2022) recovered much of the resulting quality gap via repeated denoising. Discrete diffusion LMs (Austin et al. 2021a; Li et al. 2022; Gong et al. 2023; Gulrajani and Hashimoto 2023; Lou, Meng, and Ermon 2024; Nie et al. 2025) scaled this paradigm to LLM size via bidirectional denoising of masked blocks, and block diffusion (Arriola et al. 2025) interpolates between this regime and autoregression by combining within-block bidirectional denoising with between-block causal conditioning. More recent work targets diffusion LMs as efficient generators in their own right: EfficientDLM (Fu et al. 2026b) uses position-dependent token masking; TiDAR (Liu et al. 2025) casts a single model as both drafter and verifier; and Nemotron-Labs-Diffusion (Fu et al. 2026a) introduces a tri-modal architecture unifying autoregressive, diffusion, and self-speculative decoding. The natural synthesis pursued in this work—block-diffusion models as speculative drafters, of which DFlash (Chen, Liang, and Liu 2026) is a prominent recent example—inherits the parallel-block speedup of these models while introducing, as we show in Section 4, a training objective in tension with the causal acceptance contract.

Aligning drafters with the acceptance contract. A rapidly growing literature targets the same gap we do. Zhou et al. (2024) study reverse-KL and total-variation alternatives to the standard forward-KL distillation loss for autoregressive drafters, and the recent LK losses of Samarin et al. (2026) optimize a TV-based per-token acceptance objective. These methods change the per-token *loss family* but weight all K draft positions uniformly and remain agnostic to the joint-prefix structure of the acceptance contract; they are also developed against autoregressive drafters (Medusa-style heads, EAGLE variants, MTP modules) rather than bidirectional block decoders. SpecDiff-2 of Sandler et al. (2025), targets the same block-diffusion-drafter / autoregressive-verifier mismatch via *streak-distillation*—a fine-tuning objective maximising a differentiable surrogate for the expected accepted streak under verifier-sampled teacher trajectories—paired with a test-time *self-selection acceptance* mechanism. The chain reward (Section 5.5) shares streak-distillation's animating idea but evaluates the surrogate along the teacher-forced ground-truth tokens already materialized by $\mathcal{L}_{\mathrm{CE}}$, adding only a cumulative sum and exponentiation per block—no verifier rollout (Section 6.5 shows that we can use both). A contemporaneous D-PACE (Wu et al. 2026) also targets this gap with a prefix-product accepted-length surrogate similar to the chain reward we investigate, but applies its gradient as a detached per-position weight on the standard cross-entropy rather than as an additive reward term. The present analysis is otherwise scoped entirely to training-time interventions and leaves the inference pipeline at the standard rejection-sampling contract; test-time mechanisms such as the tree-based selection of DDTree (Ringel and Romano 2026) are orthogonal and can be combined with the interventions we explore (Sections 6.4–6.5).

3 Preliminaries

3.1 Notation

We consider an autoregressive target model $p_\phi(x_t \mid x_{<t})$ over a vocabulary $\mathcal{V}$ with parameters ϕ, generating sequences $x_{1:T}$ according to $p_\phi(x_{1:T}) = \prod_{t=1}^{T} p_\phi(x_t \mid x_{<t})$. We write $h_{1:t}^{(\ell)}$ for the hidden states produced by the ℓ-th transformer block of p_ϕ when conditioned on $x_{<t}$. A *drafter* is a parametric distribution $q_\psi(x_t \mid x_{<t})$ designed to approximate p_ϕ at lower per-token cost.

3.2 Speculative Decoding

Given a prefix $x_{<t}$ and a draft horizon $K \in \mathbb{N}$, one iteration of speculative decoding (Leviathan, Kalman, and Matias 2023; Chen et al. 2023) proceeds in three phases. The drafter first samples K candidate tokens $\tilde{x}_t, \ldots, \tilde{x}_{t+K-1}$ from q_ψ. The target is then invoked once on the extended sequence and produces, in a single parallel pass, the conditionals $p_\phi(\cdot \mid x_{<t+k})$ for all $k \in \{0, \ldots, K\}$, where $x_{<t+k} \equiv (x_{<t}, \tilde{x}_{t:t+k-1})$. The candidates are finally traversed left-to-right; $\tilde{x}_{t+k}$ is accepted with probability

$$a_k \;=\; \min\!\left(1,\; \frac{p_\phi(\tilde{x}_{t+k} \mid x_{<t+k})}{q_\psi(\tilde{x}_{t+k} \mid x_{<t+k})}\right), \tag{1}$$

and on first rejection a replacement is drawn from the residual distribution

$$x_{t+k} \;\sim\; \mathrm{norm}\big(\max(0,\; p_\phi(\cdot \mid x_{<t+k}) - q_\psi(\cdot \mid x_{<t+k}))\big), \tag{2}$$

terminating the iteration. If all K candidates are accepted, one bonus token is sampled from $p_\phi(\cdot \mid x_{<t+K})$ at no extra cost. The tokens emitted by this procedure are distributed identically to those of standard autoregressive sampling from p_ϕ (Leviathan, Kalman, and Matias 2023).

Writing $\alpha_k = \mathbb{E}_{\tilde{x}}[a_k]$ for the marginal acceptance probability at position k and assuming i.i.d. rate α across positions, the expected number of tokens emitted per iteration is

$$\tau(\alpha, K) \;=\; \frac{1 - \alpha^{K+1}}{1 - \alpha}, \tag{3}$$

including the bonus token. With $c \in [0, 1]$ the wall-clock cost of one drafter pass relative to one target pass, the expected speedup over standard decoding is

$$\mathcal{S}(\alpha, K, c) \;=\; \frac{1 - \alpha^{K+1}}{(1 - \alpha)\,(1 + Kc)}. \tag{4}$$

$\mathcal{S}$ grows with both K and α, but each increment to K also incurs additive drafter cost and, typically, a decrease in α at deeper positions (Li et al. 2024a). Drafter design reduces to maximising the realised profile $\{\alpha_k\}$ subject to a constraint on c.

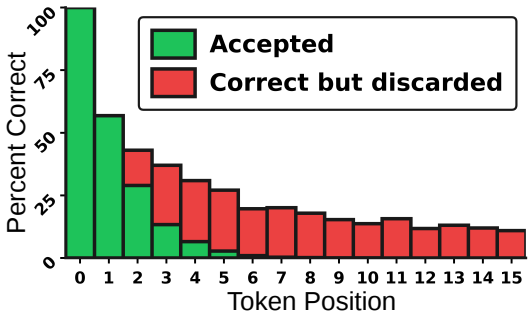

Figure 2: Per-position breakdown of the drafter's correct predictions on HumanEval, for a position-uniform DFlash baseline trained against Llama-3-8B-Instruct with block size $K = 16$. Bars are stacked: the green segment is the fraction of draft tokens accepted under the rejection-sampling contract (1), and the red segment is the fraction that actually match the target, but is nevertheless discarded as a consequence of an upstream rejection within the same block. Per-position correctness (total bar height) decays only mildly with offset k, whereas per-position acceptance (green) decays at the geometric rate implied by (3), so the discarded fraction grows monotonically and dominates beyond a short prefix.

Benchmark	Accept (%)	Correct (%)	Waste (%)
GSM8K	11.6	21.9	46.9
MT-Bench	9.9	17.3	42.9
HumanEval	13.3	27.9	52.6
AIME	12.1	23.0	47.4
MBPP	11.3	20.4	44.5
Avg.	11.6	22.1	46.9

Table 1: Draft-token utilisation of a position-uniform DFlash drafter trained against Llama-3-8B-Instruct with block size $K = 16$, evaluated across five benchmarks. Each block contributes 16 candidate slots: 15 drafted positions plus the target's bonus correction sampled at the rejection point (always retained, always matches the target's greedy output). *Accept* is the fraction of the 16 slots whose token is kept by the rejection-sampling contract (1), i.e. accepted drafts plus the bonus; *Correct* is the fraction whose draft (or bonus) matches the target's greedy output; *Waste* is the fraction of those correct tokens that the verification contract is nevertheless forced to discard due to an upstream rejection within the same block, i.e. $\text{Wasted}/(\text{Accepted} + \text{Bonus} + \text{Wasted})$.

3.3 Block-Diffusion Drafters

Discrete diffusion language models (Austin et al. 2021a; Lou, Meng, and Ermon 2024; Nie et al. 2025) replace the autoregressive factorization with a denoising formulation: a forward chain progressively masks a clean sequence, and a reverse model trained under bidirectional self-attention emits tokens by iteratively denoising a fully masked block. Block diffusion (Arriola et al. 2025) interpolates between this regime and ordinary autoregression by partitioning a sequence into contiguous blocks $\mathbf{b}_1, \ldots, \mathbf{b}_N$ of size K and factoring

$$p_\theta(x_{1:T}) \;=\; \prod_{n=1}^{N} p_\theta(\mathbf{b}_n \mid \mathbf{b}_{<n}), \tag{5}$$

where each block-conditional is parameterised by a discrete diffusion model with bidirectional attention restricted to positions inside $\mathbf{b}_n$.

A *block-diffusion drafter* instantiates a block-diffusion model as the speculative drafter q_ψ in the framework of (1)–(4); DFlash (Chen, Liang, and Liu 2026) is a prominent recent realization of this design and the concrete instance we adopt throughout. Following the feature-level drafting principle of EAGLE (Li et al. 2024b, 2025b), the drafter is conditioned not on discrete tokens but on hidden states extracted from a fixed subset of target layers $\mathcal{L} = \{\ell_1, \ldots, \ell_L\}$. Concretely, given a verified prefix $x_{<t}$ already processed by p_ϕ, the drafter consumes the multi-layer feature tensor $H_{<t} = [h_{<t}^{(\ell_1)} \,\|\, \cdots \,\|\, h_{<t}^{(\ell_L)}]$ and emits the entire K-token candidate block in a single non-autoregressive forward pass through a small denoising transformer f_ψ with full self-attention over the K draft positions:

$$(\tilde{x}_t, \ldots, \tilde{x}_{t+K-1}) \;\sim\; q_\psi(\cdot \mid H_{<t}) \;=\; \mathrm{Cat}(f_\psi(H_{<t})). \tag{6}$$

The parallel block is then fed into the target verification step unchanged. In our experiments we set $K = 16$.

Training objective. The drafter is trained by teacher-forced position-wise cross-entropy: for a target block $\mathbf{b}_n = (x_t, \ldots, x_{t+K-1})$ with corresponding features $H_{<t}$,

$$\mathcal{L}_{\mathrm{CE}}(\psi) \;=\; -\,\mathbb{E} \sum_{k=1}^{K-1} \log q_\psi(x_{t+k} \mid H_{<t}), \tag{7}$$

where each summand is computed under bidirectional self-attention over the K draft positions. The weighting in (7) is *uniform* across positions, whereas the speedup $\mathcal{S}(\alpha, K, c)$ depends on the *compounding* of left-to-right acceptance probabilities through (3). This mismatch between the symmetric training-time objective and the asymmetric verification-time reward is the central technical problem the remainder of this work addresses.

4 The Training–Verification Gap

The DFlash training objective (7) and the throughput functional (3) measure different things. The loss $\mathcal{L}_{\mathrm{CE}}$ is a position-wise sum of log-likelihoods under bidirectional self-attention, weighting all K slots equally; the gradient at position k depends on neither the value nor the acceptance of any earlier draft token. The expected accepted length, by contrast, depends on the draft as a whole: position k contributes to $\tau(\alpha, K)$ only when every preceding position is accepted under (1), and that joint event decays geometrically in k whenever per-position acceptance is below one. The training loss therefore spends capacity as if the K positions

ID	LR	γ	α_{f}	α_{c}	MT-Bench τ	GSM8K τ	HumanEval τ	AIME τ	MBPP τ	LiveCode τ	Avg. τ
a	1e-4	None	0	0	1.377	1.320	1.389	1.169	1.327	1.310	1.315
b	**1e-3**	**None**	**0**	**0**	**1.931**	**2.279**	**2.803**	**2.705**	**2.298**	**2.241**	**2.376**
c	1e-2	None	0	0	1.670	1.905	2.319	2.031	1.907	1.861	1.949
d	1e-3	7	0	0	2.050	2.289	2.949	3.017	2.239	2.232	2.463
e	**1e-3**	**10**	**0**	**0**	**2.105**	**2.382**	**3.126**	**3.152**	**2.412**	**2.317**	**2.583**
f	1e-3	20	0	0	2.017	2.218	2.921	2.812	2.318	2.219	2.418
g	**1e-3**	**10**	**0.3**	**0**	**2.222**	**2.753**	**3.521**	**3.465**	**2.720**	**2.668**	**2.892**
h	1e-3	10	0.5	0	2.056	2.650	3.519	3.678	2.637	2.551	2.849
i	1e-3	10	1	0	2.204	2.650	3.401	3.648	2.606	2.487	2.833
k	1e-3	10	0.3	5	2.222	2.703	3.724	3.833	2.664	2.691	2.973
j	1e-3	10	0.3	10	2.185	2.762	3.957	4.278	2.904	2.769	3.143
l	1e-3	10	0.3	20	2.214	2.940	3.944	4.201	2.753	2.825	3.146
m	1e-3	10	0.3	30	2.391	2.970	4.223	4.734	3.032	2.843	3.365
n	**1e-3**	**10**	**0.3**	**40**	**2.341**	**3.074**	**4.336**	**4.757**	**3.088**	**2.922**	**3.420**
o	1e-3	10	0.3	50	2.349	3.044	4.192	4.607	3.023	2.934	3.358

Table 2: Average acceptance length (τ) across evaluation benchmarks for different training configurations. γ is the loss-decay constant; α_{f} is the first-error focal coefficient; and α_{c} is the chain-loss coefficient. The draft horizon is fixed throughout. Higher τ is better. Bold entries indicate the best-performing configuration within each ablation group.

contributed independently to throughput, while the verifier credits only the longest causally consistent prefix. An early-position error truncates the rest of the block; an error deep in the block typically has no effect at all.

Figure 2 quantifies the resulting waste at the individual-token level. For a standard block-diffusion drafter (DFlash trained against Llama-3-8B-Instruct and evaluated on HumanEval), draft tokens that match the target's greedy output are split by block position into those that verification accepts (green) and those discarded because of an upstream rejection in the same block (red). Per-position correctness decays only mildly with k, consistent with the symmetric training signal; per-position acceptance decays at the geometric rate implied by (3), so the red region grows steadily with k and dominates beyond a short prefix.

Table 1 confirms that the effect persists across workloads: across GSM8K, MT-Bench, HumanEval, AIME, and MBPP, the drafter's proposals match the target's greedy output in 22.1% of block slots on average, but only 11.6% of slots survive the rejection-sampling contract, so an average of 46.9% of the drafter's correct predictions are discarded as a consequence of upstream rejections, with per-benchmark waste rates between 42.9% and 52.6%.

Together, Figure 2 and Table 1 pinpoint how capacity is misallocated. The loss (7) keeps rewarding likelihood gains at deep positions whose expected contribution to (3) is geometrically suppressed, while underweighting gains at shallow positions whose acceptance gates the entire suffix. The rest of this work studies modifications to the training signal that bring the per-position weight of $\mathcal{L}_{\mathrm{CE}}$ closer to each position's actual contribution to the verifier's accepted length.

5 Training Techniques to Bridge the Gap

The three interventions studied in this section reshape the drafter's training loss along orthogonal axes. Loss decay (Section 5.3) reweights the per-position contribution to $\mathcal{L}_{\mathrm{CE}}$ along the *position* axis as a hand-specified, ex-ante function of k. The first-error focal loss (Section 5.4) is sparser: it adds an auxiliary cross-entropy term restricted to the single position per block that the drafter's argmax decoder currently mispredicts—the *chain breaker* whose rejection truncates the rest of the block—and contributes nothing on blocks the drafter already decodes correctly. The chain reward (Section 5.5) is the densest of the three along the *joint-prefix* axis: it augments the loss with a differentiable surrogate for the expected accepted length, so that the gradient at every position is reweighted online by the drafter's current estimate of the prefix-acceptance probability. The three interventions are mutually composable, and the experiments below layer them in this order.

The peak learning rate is ablated first to factor optimizer dynamics out of the subsequent loss-shape comparisons. Table 2 reports the average acceptance length τ on the standard suite for every configuration and is referenced throughout this section; Table 3 extends the same configurations across Llama-3.2-3B, Llama-3-8B, Qwen-3-4B, and Qwen-3-8B as a cross-target generalisation check.

5.1 Experimental Setup

Target. All ablations in Table 2 use Meta-Llama-3-8B-Instruct as the target p_ϕ: 32 transformer blocks, hidden size 4096, 32 attention heads with 8 KV heads (grouped-query attention), intermediate size $14{,}336$, vocabulary size $128{,}256$, RoPE base 5×10^5 with the Llama-3 long-context scaling.

Drafter. The drafter q_ψ is a four-layer DFlash denoiser of the architecture introduced in Section 3: four full-attention transformer blocks, hidden size and head configuration matched to the target ($d = 4096$, 32:8 GQA, $d_{\mathrm{head}} = 128$, $d_{\mathrm{ff}} = 14{,}336$), bfloat16, block size $K = 16$. Multi-layer feature conditioning $\mathcal{L}$ is set to the four evenly-spaced target layers $\{0, 10, 20, 30\}$.

Dataset. Training data is ShareGPT (Aeala, gozfarb, and

anon8231489123 2023): a compilation of multi-turn user–assistant conversations rendered into Llama-3 chat-template format and truncated at a sequence length of 4,096 tokens. Within each sequence, 512 block anchors are sampled per step to construct $(H_{<t}, \mathbf{b}_n)$ training pairs as defined in (6). The corpus is not target-distilled: prompts and assistant continuations are taken as-is.

Optimization. Drafters are trained for 3 epochs with AdamW (Loshchilov and Hutter 2019) under a cosine schedule, a warm-up fraction of 1.5%, and a per-device micro batch of 1 with gradient accumulation of 4, yielding an effective batch size of 32 across the 8 NVIDIA H100 80 GB GPUs (NVIDIA Corporation 2022) of a single node (1-way tensor parallel, pure data parallel). The peak learning rate is the value reported in each row of Table 2; unless mentioned, peak LR is fixed at the value selected in Section 5.2. Training is done using the SpecForge framework (Li et al. 2025a).

Evaluation. Acceptance length τ is averaged over six benchmarks spanning open-ended dialogue (MT-Bench; Zheng et al. 2023), mathematical reasoning (GSM8K, Cobbe et al. 2021; AIME, drawn from the AI-MO validation set of 90 problems from AIME 2022–2024, Project Numina (AI-MO) 2024), and code generation (HumanEval, Chen et al. 2021; MBPP, Austin et al. 2021b; LiveCodeBench, Jain et al. 2024), reported per-benchmark and as the mean $\bar{\tau}$ in the rightmost column of Table 2.

5.2 Impact of Learning Rate

The peak learning rate of the AdamW optimizer is ablated first with all other hyperparameters held fixed at the configuration of Section 5.1. Rows a–c of Table 2 sweep the peak learning rate over $\{10^{-4}, 10^{-3}, 10^{-2}\}$ at the position-uniform baseline ($\gamma = \texttt{None}$, $\alpha_c = \alpha_f = 0$). The optimum sits at 10^{-3} (row b), which yields $\bar{\tau} = 2.376$ against 1.315 at 10^{-4} (row a) and 1.949 at 10^{-2} (row c)—i.e. both an order of magnitude lower and an order of magnitude higher cost the drafter 45% and 18% of average acceptance length, respectively. All subsequent experiments fix the peak learning rate at 10^{-3}.

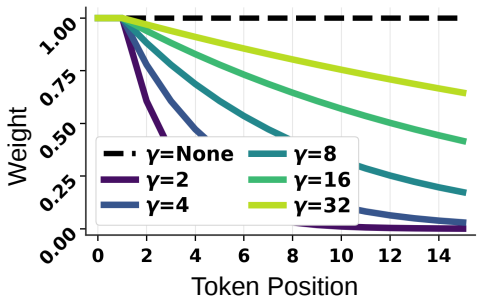

Figure 3: Per-position loss weight $w_k(\gamma)$ of (8) as a function of the within-block token offset $k \in \{0, \ldots, K-1\}$, plotted for several decay constants γ at block size $K = 16$.

5.3 Position-Wise Reweighting: Loss Decay

The simplest reconciliation between the position-uniform weight implicit in (7) and the geometrically decaying inference value of each position, established in Section 4, is to promote the per-position weight to an explicit hyperparameter. This parameterisation was introduced in the original DFlash work (Chen, Liang, and Liu 2026); we adopt it here and study it more systematically as one of three composable interventions. The per-position weight at offset $k \in \{1, \ldots, K-1\}$ (the anchor $k = 0$ is excluded from the loss, as it is in (7)) is set to

$$w_k(\gamma) \;=\; \exp\!\left(-\frac{k-1}{\gamma}\right), \tag{8}$$

parameterised by a decay constant $\gamma > 0$. The reweighted objective then reads

$$\mathcal{L}_{\mathrm{CE}}^{(\gamma)}(\psi) \;=\; -\,\mathbb{E}\sum_{k=1}^{K-1} w_k(\gamma)\,\log q_\psi(x_{t+k} \mid H_{<t}). \tag{9}$$

The limit $\gamma \to \infty$ recovers the position-uniform objective of (7); the limit $\gamma \to 0^+$ collapses to a single-position objective at $k = 1$. Intermediate values induce an exponential profile that, by construction, allocates more gradient signal to the early positions whose acceptance gates the entire suffix and less to the deep positions whose contribution to (3) is geometrically suppressed.

This is the cheapest intervention available: it adds no forward pass through either the drafter or the target, no architectural change, and a single scalar hyperparameter. Its principal limitation is that the weighting profile (8) is fixed and decoupled from the drafter's per-position acceptance behavior at any point in training; every block is downweighted identically, irrespective of its difficulty or of which positions the current drafter is already capable of resolving. Sections 5.4 and 5.5 address this limitation in complementary ways: the first-error focal loss conditions on the chain-breaker position selected by the current drafter, and the chain reward couples the per-position weight to the drafter's own prefix-acceptance probability.

Rows d–f of Table 2 sweep $\gamma \in \{7, 10, 20\}$ at the tuned learning rate of Section 5.2; the position-uniform baseline (row b) records $\bar{\tau} = 2.376$ as the reference point. Performance peaks at the intermediate setting: $\gamma = 7$ yields $\bar{\tau} = 2.463$ (+3.7%) and $\gamma = 20$ yields $\bar{\tau} = 2.418$ (+1.8%), while $\gamma = 10$ (row e) reaches $\bar{\tau} = 2.583$, a +8.7% gain over the position-uniform baseline. This matches the reasoning of Section 4: a sharp profile ($\gamma = 7$) cuts off the deep-position signal too aggressively, a slack profile ($\gamma = 20$) is close to the uniform baseline, and the intermediate setting best tracks the geometrically decaying inference value of each position implied by (3). We fix $\gamma = 10$ for subsequent ablations.

5.4 Targeting the Chain Breaker: Focal Loss

The decay weighting of Section 5.3 reshapes the gradient with a position-only schedule applied uniformly to every block. The analysis of Section 4 suggests a more targeted intervention. Under the rejection-sampling contract (1), a block of K draft tokens contributes to the accepted length up to and

Model	γ	α_{f}	α_{c}	MT-Bench τ	GSM8K τ	HumanEval τ	AIME τ	MBPP τ	LiveCode τ	Avg. τ
Llama 3.2-3B	None	0	0	2.351	2.875	3.778	3.003	3.110	2.546	2.944
	10	0	0	2.454	3.040	4.096	3.356	3.300	2.636	3.147
	10	0.3	0	2.542	3.110	4.347	3.455	3.382	2.681	3.253
	10	**0.3**	**40**	**2.717**	**3.507**	**5.196**	**4.197**	**3.933**	**2.975**	**3.754**
Llama-3-8B	None	0	0	1.931	2.279	2.803	2.705	2.298	2.241	2.376
	10	0	0	2.105	2.382	3.126	3.152	2.412	2.317	2.583
	10	0.3	0	2.222	2.753	3.521	3.465	2.720	2.668	2.892
	10	**0.3**	**40**	**2.341**	**3.074**	**4.336**	**4.757**	**3.088**	**2.922**	**3.420**
Qwen3-4B	None	0	0	1.875	2.397	2.738	2.133	2.396	2.265	2.301
	10	0	0	1.959	2.502	3.002	2.253	2.596	2.387	2.450
	10	0.3	0	2.000	2.598	3.126	2.357	2.618	2.481	2.530
	10	**0.3**	**40**	**2.123**	**2.833**	**3.507**	**2.554**	**2.958**	**2.713**	**2.781**
Qwen3-8B	None	0	0	1.836	2.248	2.480	2.030	2.032	2.144	2.128
	10	0	0	1.901	2.517	2.779	2.233	2.241	2.315	2.331
	10	0.3	0	1.961	2.608	2.969	2.351	2.306	2.420	2.436
	10	**0.3**	**40**	**2.055**	**2.795**	**3.384**	**2.558**	**2.466**	**2.665**	**2.654**

Table 3: Acceptance length (τ) across benchmarks for different target-model families and training configurations. γ is the loss-decay constant (`None` denotes the position-uniform objective), α_{f} is the first-error focal coefficient, and α_{c} is the chain-loss coefficient. Higher τ is better.

including the first position at which the draft is rejected; the identity of that single position—the *chain breaker*—is sufficient to determine the block's acceptance outcome under the argmax simplification. From the perspective of the throughput functional (3), fixing the chain breaker of a block whose prefix is otherwise correct extends its accepted length by exactly one, whereas additional gradient on positions deeper than the breaker yields no improvement until the breaker itself is fixed.

The first-error focal loss instantiates this intuition as an auxiliary cross-entropy term restricted to the chain-breaker position of each block. Concretely, for each block n with draft logits $\ell_{n,k}$ and ground-truth tokens $x^*_{n,k}$, denote the argmax decoder predictions by $\hat{x}_{n,k} = \arg\max_v \ell^v_{n,k}$ and the set of positions where the argmax disagrees with the gold by $E_n = \{k \in \{1, \dots, K-1\} : \hat{x}_{n,k} \neq x^*_{n,k}\}$. Blocks with $E_n = \emptyset$ contribute nothing to the auxiliary term. For blocks with at least one disagreement, let $k^*_n = \min E_n$ be the chain-breaker offset and define

$$\mathcal{L}_{\mathrm{fe}}(\psi) \;=\; \frac{\sum_{n:\, E_n \neq \emptyset} w_{k^*_n}(\gamma)\, \big(-\log q_\psi(x^*_{n,k^*_n} \mid H_{<t_n})\big)}{\sum_{n:\, E_n \neq \emptyset} w_{k^*_n}(\gamma) \;+\; \varepsilon}, \tag{10}$$

the mean per-position cross entropy at the chain breaker, weighted by the same position decay $w_k(\gamma)$ used in (9). With $\alpha_{\mathrm{f}} \geq 0$ the focal coefficient, the decay-plus-focal objective reads

$$\mathcal{L}^{(\gamma,\alpha_{\mathrm{f}})}(\psi) \;=\; \mathcal{L}^{(\gamma)}_{\mathrm{CE}}(\psi) \;+\; \alpha_{\mathrm{f}}\, \mathcal{L}_{\mathrm{fe}}(\psi). \tag{11}$$

The construction is qualitatively distinct from the decay weighting. Loss decay applies at every position of every block, redistributing gradient by a smooth function of k. The first-error focal term, by contrast, is *block-conditional* (it activates only when the block is breakable) and *position-sparse* (it activates at exactly one offset per such block, selected adaptively by the drafter's current decoder). Two further consequences follow. First, blocks that the drafter already decodes correctly receive no extra signal, which prevents the auxiliary term from over-fitting easy patterns. Second, as the drafter improves, the chain-breaker distribution shifts deeper into the block, so the auxiliary term automatically retargets toward later positions without an explicit schedule.

Rows g–i of Table 2 sweep $\alpha_{\mathrm{f}} \in \{0.3, 0.5, 1.0\}$ on top of the tuned $\gamma = 10$ of Section 5.3 (row e, $\bar{\tau} = 2.583$). All three settings improve over the decay-only baseline, and the objective is monotone-decreasing in α_{f} over this range: $\alpha_{\mathrm{f}} = 0.3$ (row g) attains $\bar{\tau} = 2.892$, a $+12.0\%$ gain over the focal-free decay baseline and a cumulative $+21.7\%$ over the position-uniform row b; $\alpha_{\mathrm{f}} = 0.5$ (row h) and $\alpha_{\mathrm{f}} = 1.0$ (row i) regress slightly to $\bar{\tau} = 2.849$ and $\bar{\tau} = 2.833$ respectively. The improvement is largest on the reasoning- and code-heavy benchmarks—HumanEval ($2.803 \to 3.521$, $+25.6\%$) and AIME ($2.705 \to 3.465$, $+28.1\%$) relative to row b. We adopt $\alpha_{\mathrm{f}} = 0.3$ as the reference setting for the chain sweep that follows.

5.5 Joint-Prefix Reweighting: Chain Reward

The decay weighting of Section 5.3 is a position-only schedule fixed ex ante; the first-error focal loss of Section 5.4 is sparse and block-conditional. A third axis of intervention, complementary to both, is to substitute a differentiable surrogate for the expected accepted length itself, so that the gradient at every position is reweighted online by the prefix-acceptance probability under the current drafter. Let $p_k = q_\psi(x_{t+k} \mid H_{<t})$ denote the drafter's marginal probability on the ground-truth token at offset k, and write $\rho_k = \prod_{j=1}^{k} p_j$ for the corresponding prefix probability.

Treating ρ_k as a proxy for the probability that positions $1, \dots, k$ of the draft are jointly accepted under (1), the ex-

ID	Technique	Setting	Avg. τ
a	Position-uniform baseline	—	2.376
b	Loss decay only	$\gamma = 10$	2.583
c	Focal only	$\alpha_{\mathrm{f}} = 0.3$	2.736
d	Chain only	$\alpha_{\mathrm{c}} = 40$	2.919

Table 4: Each training intervention applied in isolation to the position-uniform baseline at the tuned learning rate of Section 5.2. Row a reproduces row b of Table 2; row b reproduces row e; rows c and d are single-technique runs with the other two coefficients held at zero. Higher τ is better.

γ	α_{f}	α_{c}	Avg. τ
None	0	0	3.914
10	0	0	4.230
10	0.3	0	4.633
10	**0.3**	**40**	**4.985**

Table 5: Average acceptance length (τ) for the four reference training configurations on the target-aligned Nemotron-V2 + CodeAlpaca split. Higher is better; the best row is bolded. All runs use the same drafter architecture, optimiser settings, and evaluation suite as Table 2.

pected accepted length is approximated by

$$R_{\mathrm{chain}}(\psi) \;=\; \frac{1}{K-1}\sum_{k=1}^{K-1}\exp\left(\sum_{j=1}^{k}\log p_j\right), \qquad (12)$$

normalised to $[0, 1]$. The full training objective combining the three interventions of Sections 5.3–5.5 is then

$$\mathcal{L}(\psi) \;=\; \mathcal{L}_{\mathrm{CE}}^{(\gamma)}(\psi) \;+\; \alpha_{\mathrm{f}}\,\mathcal{L}_{\mathrm{fe}}(\psi) \;-\; \alpha_{\mathrm{c}}\,R_{\mathrm{chain}}(\psi), \quad (13)$$

with $\alpha_{\mathrm{c}} \geq 0$ controlling the trade-off between marginal correctness and joint chain probability. Differentiating (12) with respect to $\log p_k$ yields $(K-1)^{-1}\sum_{j\geq k}\rho_j$: each position is reinforced in proportion to the cumulative prefix probability of its own and every deeper position, so that early positions receive a strictly larger gradient than deep ones, with the relative weighting determined by the drafter's current calibration rather than by a hand-specified schedule.

The implementation reuses the per-token log-probabilities already computed for $\mathcal{L}_{\mathrm{CE}}$, since $-\mathrm{CE}(\mathrm{logits}, x^*) = \log p$; the chain reward therefore adds only a cumulative sum and exponentiation per block and does not require a second pass through the language modelling head.

Rows k–o of Table 2 sweep $\alpha_{\mathrm{c}} \in \{5, 10, 20, 30, 40, 50\}$ on top of the $(\gamma, \alpha_{\mathrm{f}}) = (10, 0.3)$ stack of Section 5.4 (row g, $\bar{\tau} = 2.892$). Even modest coefficients are markedly accretive: $\alpha_{\mathrm{c}} = 5$ (row k) raises $\bar{\tau}$ to 2.973; $\alpha_{\mathrm{c}} = 10$ (row j) to 3.143; $\alpha_{\mathrm{c}} = 20$ (row l) to 3.146; $\alpha_{\mathrm{c}} = 30$ (row m) to 3.365; and $\alpha_{\mathrm{c}} = 40$ (row n) peaks at $\bar{\tau} = 3.420$, a $+18.3\%$ gain over the chain-free row g and a cumulative $+43.9\%$ over the position-uniform baseline of row b. The improvement is largest on the reasoning- and code-heavy benchmarks—HumanEval ($2.803 \rightarrow 4.336$, $+54.7\%$) and AIME ($2.705 \rightarrow 4.757$, $+75.9\%$) relative to row b—where long, well-determined draft chains are most plentiful and most rewarded by the joint-prefix reweighting. The objective turns over past the peak: at $\alpha_{\mathrm{c}} = 50$ (row o) $\bar{\tau}$ regresses to 3.358, as the chain term begins to dominate the per-position cross entropy. We adopt $\alpha_{\mathrm{c}} = 40$ as the reference setting for the cross-target generalisation experiments that follow.

5.6 Experiments on Other Models

Table 3 replicates the four reference configurations of Section 5.1—position-uniform baseline ($\gamma = \texttt{None}, \alpha_{\mathrm{f}} = \alpha_{\mathrm{c}} = 0$), $+\gamma$, $+\alpha_{\mathrm{f}}$, $+\alpha_{\mathrm{c}}$—across four instruction-tuned target models: Llama-3.2-3B-Instruct (Meta AI 2024), Meta-Llama-3-8B-Instruct (Grattafiori et al. 2024), and the instruction-tuned Qwen3-4B and Qwen3-8B releases (Yang et al. 2025) — to confirm that the ranking is target-agnostic. All four are the instruction-tuned variants of their respective families; no separate post-training is performed on the target. The compounding pattern of Table 2 reproduces on every target: each successive intervention strictly improves average acceptance length, and the fully stacked $(\gamma, \alpha_{\mathrm{f}}, \alpha_{\mathrm{c}}) = (10, 0.3, 40)$ configuration delivers the largest $\bar{\tau}$ for all four targets, with cumulative gains over the position-uniform baseline of $+27.5\%$ (Llama-3.2-3B), $+43.9\%$ (Llama-3-8B), $+20.9\%$ (Qwen-3-4B), and $+24.7\%$ (Qwen-3-8B). The absolute gains are largest on the reasoning- and code-heavy benchmarks (HumanEval, AIME) across every target, mirroring the per-benchmark profile of Section 5.5 and indicating that the joint-prefix reweighting transfers target-independently to the workloads on which long, well-determined draft chains are most plentiful.

6 Ablations

6.1 Each Technique in Isolation

The sweeps of Section 5 layer the three interventions cumulatively: loss decay is tuned alone, the focal term is added on top of the chosen γ, and the chain reward is added on top of the decay–plus–focal stack. This construction is the one that yields the strongest configuration of Table 2, but it leaves open the question of how much each intervention contributes *in isolation*—i.e., applied directly to the position-uniform baseline of (7) with the other two coefficients held at zero. We isolate each technique by holding the learning rate fixed at the value selected in Section 5.2 and training a separate drafter for each of the four single-technique configurations.

Three observations follow from Table 4. First, every intervention is individually accretive over the position-uniform baseline, and the relative ranking (chain > focal > decay) is consistent with the analysis of Section 4: the more directly an intervention couples its gradient to the verification rule, the larger its isolated effect. Loss decay reweights by a fixed ex-ante schedule that only approximately tracks the geometric decay of acceptance value; the focal term conditions on the drafter's own argmax breaker; the chain reward integrates the joint prefix probability across all positions. Second, the chain reward alone already recovers a large fraction of the fully-

Block	Base			Aligned		
	AIME τ	HumanEval τ	MBPP τ	AIME τ	HumanEval τ	MBPP τ
2	1.725	1.734	1.605	1.876	1.840	1.720
4	2.462	2.471	2.118	3.238	3.003	2.475
8	2.654	2.561	2.278	4.309	3.919	2.945
16	**2.705**	**2.803**	**2.298**	**4.757**	**4.336**	**3.088**
32	2.656	2.750	2.312	4.577	4.069	3.004

Table 6: Acceptance length (τ) as a function of the speculation horizon K for the position-uniform baseline (Base) and the fully-aligned $(\gamma, \alpha_{\mathrm{f}}, \alpha_{\mathrm{c}}) = (10, 0.3, 40)$ drafter (Aligned). Drafters are trained at $K = 16$ and evaluated at the indicated horizon. Higher is better; the best row per column is bolded.

Technique	No ddTree		With ddTree	
	Avg. TPS	Avg. τ	Avg. TPS	Avg. τ
Uniform Decay	126.75	2.376	188.68	3.791
+ Gamma Decay	142.97	2.583	204.96	3.999
+ Focal Loss	168.95	2.892	238.81	4.328
+ Chain Loss	225.12	3.420	294.68	4.609

Table 7: Average throughput (tokens per second, TPS) and average accepted-token length (τ) under progressive training enhancements, comparing standard speculative decoding (no ddTree, contiguous block acceptance with block size 16) and ddTree verification. Results are averaged across six benchmarks (GSM8K, HumanEval, AIME25, MBPP, MT-Bench, and LiveCodeBench). Higher is better.

stacked gain (row n of Table 2, $\bar{\tau} = 3.420$), suggesting that the joint-prefix axis is the dominant source of the improvement. Third, the gap between the chain-only configuration and the fully-stacked one nevertheless remains substantial, suggesting that the three interventions do not encode the same training signal.

6.2 Impact of Training Data

All experiments of Section 5 use the ShareGPT dataset (Aeala, gozfarb, and anon8231489123 2023) as the drafter training corpus, taken as-authored rather than re-distilled from the target. To characterize how data composition interacts with the loss-shape interventions of Section 5, we re-run the four reference configurations on a target-aligned corpus. To provide a diverse mixture of instruction-following and code prompts, following (Chen, Liang, and Liu 2026), we collect approximately 800K samples from the NVIDIA Nemotron Post-Training Dataset V2 (Nathawani and NVIDIA 2025) and CodeAlpaca (Chaudhary 2023). Rather than training against the as-authored assistant responses, we regenerate each completion under the target model p_ϕ (Meta-Llama-3-8B-Instruct, Grattafiori et al. 2024) served via sglang at temperature 0.7, `max_tokens` $= 2048$, and bfloat16, yielding a target-aligned training split of the same prompt set. The four reference configurations of Section 5 are then retrained on this split for six epochs at the tuned learning rate and effective batch size of Section 5.1; metrics are reported at the $75{,}000$-step checkpoint.

Two observations follow from Table 5. First, the relative ordering and the cumulative compounding pattern of Section 5 both transfer to the target-aligned data: each successive intervention adds to $\bar{\tau}$ (decay $+8.1\%$, focal a further $+9.5\%$, chain a further $+7.6\%$, for a cumulative $+27.4\%$ over the position-uniform baseline), and the fully stacked configuration remains strictly best. Second, the change of training corpus is itself a large independent lever: the position-uniform baseline rises from $\bar{\tau} = 2.376$ on ShareGPT (row b of Table 2) to $\bar{\tau} = 3.914$ on the target-aligned split, a $+64.7\%$ absolute lift before any of the loss-shape interventions are applied; composed with them, the fully stacked configuration reaches $\bar{\tau} = 4.985$, a $+109.8\%$ improvement over the ShareGPT position-uniform baseline. The training-time interventions of Section 5 therefore stack multiplicatively with target-aligned training data rather than substituting for it.

6.3 Impact of Block Size for Evaluation

The draft horizon K controls both the per-iteration speedup ceiling (4) and the geometric decay of the per-position acceptance profile $\{\alpha_k\}$. To characterise this trade-off, we evaluate the position-uniform baseline (row b of Table 2) and the fully-aligned configuration (row n) at speculation horizons $K \in \{2, 4, 8, 16, 32\}$, holding training at the $K = 16$ block size of Section 5.1 fixed. Table 6 reports the average acceptance length on AIME, HumanEval, and MBPP.

Two patterns emerge. First, $\bar{\tau}$ peaks at the training block size of 16 for both drafters and falls off once the inference horizon exceeds it, with $K = 32$ regressing on every benchmark. The drafter was trained to predict 16-token blocks under bidirectional attention; pushing the inference horizon past that size asks the model to predict positions it never saw during training, and the resulting acceptance loss outweighs the extra speedup ceiling. Second, the aligned drafter's gain over the baseline grows steadily with K below the training horizon: at $K = 2$ the aligned drafter improves AIME $\bar{\tau}$ by $+8.8\%$, at $K = 4$ by $+31.5\%$, at $K = 8$ by $+62.4\%$, and at $K = 16$ by $+72.0\%$ (with comparable trajectories on HumanEval and MBPP). The position-uniform baseline stops improving well before $K = 16$ because the deep-position acceptance rate is already near zero; the aligned drafter, whose training signal explicitly targets the early-position acceptance gates and the joint prefix probability, keeps gaining speedup all the way up to the training block size.

6.4 Application to Inference-time Strategies

The training-time interventions of Section 5 leave the inference pipeline at the standard rejection-sampling contract: each round verifies a single drafted trajectory of K tokens. A complementary test-time mechanism is to exploit the fact that a single block-diffusion drafter pass already produces a per-position distribution $q_i(\cdot \mid H_{<t})$ at every offset, rather than only the argmax sample. DDTree (Ringel and Romano 2026) converts these marginals into a draft tree of at most B nodes whose prefix probabilities under the factorised distribution $Q = \prod_i q_i$ are recovered by a best-first heap walk, and verifies the resulting tree in one target-model forward pass with an ancestor-only attention mask. DDTree is wholly orthogonal to the training-time interventions studied here: it changes neither the drafter objective nor the rejection-

Technique	LiveCodeBench τ	AIME25 τ	HumanEval τ
Uniform Decay	2.263	2.262	2.493
+ Gamma Decay	2.621	2.824	2.928
+ Focal Loss	2.895	3.425	3.269
+ Chain Loss	3.361	4.650	4.202

Table 8: Average accepted-token length (τ) on LiveCodeBench, AIME25, and HumanEval for SpecDiff2-trained draft models under progressive training enhancements. Results are reported using the final checkpoint of each training run. Higher is better.

sampling exactness contract, only the set of continuations the verifier scores per round.

Table 7 stacks DDTree verification on top of the four reference configurations of Section 5 and reports both throughput and acceptance length averaged across the six benchmarks. The two axes compose cleanly. Holding the drafter at the position-uniform baseline, DDTree alone raises $\bar{\tau}$ from 2.376 to 3.791 ($+59.6\%$) and average TPS from 126.75 to 188.68 ($+48.9\%$). Holding inference at standard contiguous-block verification, the layered training-time stack raises $\bar{\tau}$ from 2.376 to 3.420 ($+44.0\%$) and TPS from 126.75 to 225.12 ($+77.6\%$). Applied jointly, the fully stacked configuration with DDTree verification reaches $\bar{\tau} = 4.609$ and 294.68 TPS, a $+94.0\%$ and $+132.5\%$ lift over the position-uniform, no-DDTree baseline. Both directions of the cross-product are monotone: every training-time increment retains its gain under DDTree, and DDTree's lift is preserved at every point along the training-time stack. The two surfaces therefore add rather than substitute, consistent with the observation that DDTree reshapes the verifier's search over fixed drafter marginals while the training-time interventions reshape the marginals themselves.

6.5 Compatibility with SpecDiff-2

The contemporaneous SpecDiff-2 of Sandler et al. (2025) targets the same training–verification mismatch we analyse through a different mechanism. Its train-time component, *streak-distillation*, replaces the position-uniform cross-entropy with a pathwise surrogate for the expected accepted streak length: at each prefix s, a continuation $x_{1:\gamma}$ is sampled from the frozen verifier p_ϕ, and the drafter's per-position marginals are scored as $\sum_{m=1}^{\gamma} \prod_{j=1}^{m} q_j(x_j \mid s)$, which is then maximised through the drafter parameters. The outer expectation is taken over teacher prefixes drawn from p_ϕ, so each gradient step requires a fresh sample of length γ from the verifier in addition to the standard drafter pass. The dominant incremental cost of streak-distillation, then, is not the surrogate itself but the verifier rollout that supplies the teacher continuation: training compute scales linearly with the rollout length and, in the published implementation, requires a verifier forward pass per training example above the drafter forward pass already needed for the base loss.

Our three training-time interventions are compatible with streak-distillation along this axis. The chain reward of Section 5.5 is also a differentiable surrogate for $\mathbb{E}[\text{accepted length}]$, but it is evaluated along the teacher-forced ground-truth tokens already materialised by $\mathcal{L}_{\text{CE}}$, so adding it costs only a cumulative sum and exponentiation per block. The decay weighting of Section 5.3 and the focal term of Section 5.4 reuse the same per-token cross-entropy tensor for the same reason. Layering the three interventions on top of streak-distillation therefore inherits the latter's rollout cost but adds nothing further, while reshaping the objective along the position and block-conditional first-error axes that streak-distillation does not address. Table 8 reports the result of this layering on LiveCodeBench, AIME25, and HumanEval, using SpecDiff-2-trained drafters as the base in every row. Streak-distillation alone (the position-uniform row of the table) records an average $\bar{\tau} = 2.339$ on this three-benchmark slice; adding $\gamma = 10$ on top raises $\bar{\tau}$ to 2.791, a $+19.3\%$ gain over the streak-distilled baseline; the $\alpha_{\text{f}} = 0.3$ focal term adds a further $+14.5\%$ for $\bar{\tau} = 3.196$ ($+36.6\%$ cumulative); and the $\alpha_{\text{c}} = 40$ chain reward adds another $+27.4\%$ for a fully stacked $\bar{\tau} = 4.071$, a $+74.0\%$ improvement over the streak-distilled baseline.

7 Conclusion

We studied the training of block-diffusion drafters for speculative decoding through the lens of the gap between the position-uniform, bidirectional objective inherited from diffusion language modelling and the strictly causal, prefix-truncating acceptance contract that governs verification. A consequence of this gap, visible in the position-uniform DFlash baseline, is that a large majority of the tokens the target ratifies as correct are discarded by the rejection-sampling rule simply because they sit downstream of an earlier within-block rejection—capacity that the standard cross-entropy loss has no mechanism to reclaim.

We analysed three complementary training-time interventions that narrow this gap—position-wise loss decay, a first-error focal loss targeting the chain-breaking position of each block, and a chain reward that substitutes a differentiable surrogate for the expected accepted length. The three reshape the loss along orthogonal axes (position, block-conditional first error, joint prefix), compose additively, preserve the exactness contract of speculative decoding, and add negligible compute over the base objective. They are individually accretive and jointly compounding across the model families and benchmark categories we evaluated, and remain orthogonal to test-time alignment mechanisms developed for the same drafter class; combining them with such mechanisms is a natural direction for follow-up work.

References

Aeala; gozfarb; and anon8231489123. 2023. ShareGPT_Vicuna_unfiltered: A cleaned dump of multi-turn user–ChatGPT conversations. https://huggingface.co/datasets/Aeala/ShareGPT_Vicuna_unfiltered.

Agrawal, A.; Panwar, A.; Mohan, J.; Kwatra, N.; Gulavani, B. S.; and Ramjee, R. 2023. SARATHI: Efficient LLM Inference by Piggybacking Decodes with Chunked Prefills. *arXiv preprint arXiv:2308.16369*.

Ankner, Z.; Parthasarathy, R.; Nrusimha, A.; Rinard, C.; Ragan-Kelley, J.; and Brandon, W. 2024. Hydra:

Sequentially-Dependent Draft Heads for Medusa Decoding. In *Conference on Language Modeling (COLM)*.

Arriola, M.; Gokaslan, A.; Chiu, J. T.; Yang, Z.; Qi, Z.; Han, J.; Sahoo, S. S.; and Kuleshov, V. 2025. Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models.

Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and van den Berg, R. 2021a. Structured Denoising Diffusion Models in Discrete State-Spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Austin, J.; Odena, A.; Nye, M.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C.; Terry, M.; Le, Q.; and Sutton, C. 2021b. Program Synthesis with Large Language Models. *arXiv preprint arXiv:2108.07732*.

Cai, T.; Li, Y.; Geng, Z.; Peng, H.; Lee, J. D.; Chen, D.; and Dao, T. 2024. Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads. In *International Conference on Machine Learning (ICML)*.

Chaudhary, S. 2023. Code Alpaca: An Instruction-following LLaMA Model for Code Generation. https://github.com/sahil280114/codealpaca.

Chen, C.; Borgeaud, S.; Irving, G.; Lespiau, J.-B.; Sifre, L.; and Jumper, J. 2023. Accelerating Large Language Model Decoding with Speculative Sampling. *arXiv preprint arXiv:2302.01318*.

Chen, J.; Liang, Y.; and Liu, Z. 2026. DFlash: Block Diffusion for Flash Speculative Decoding. In *International Conference on Machine Learning (ICML)*.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. d. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating Large Language Models Trained on Code. *arXiv preprint arXiv:2107.03374*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Dao, T. 2024. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *International Conference on Learning Representations (ICLR)*.

Dao, T.; Fu, D. Y.; Ermon, S.; Rudra, A.; and Ré, C. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.

DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*.

Dettmers, T.; Lewis, M.; Belkada, Y.; and Zettlemoyer, L. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale.

Du, C.; Jiang, J.; Yuanchen, X.; Wu, J.; Yu, S.; Li, Y.; Li, S.; Xu, K.; Nie, L.; Tu, Z.; and You, Y. 2024. GliDe with a CaPE: A Low-Hassle Method to Accelerate Speculative Decoding. In *International Conference on Machine Learning (ICML)*.

Frantar, E.; and Alistarh, D. 2023. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. In *International Conference on Machine Learning (ICML)*.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2023. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. In *International Conference on Learning Representations (ICLR)*.

Fu, Y.; Bailis, P.; Stoica, I.; and Zhang, H. 2024. Break the Sequential Dependency of LLM Inference Using Lookahead Decoding. In *International Conference on Machine Learning (ICML)*.

Fu, Y.; Whalen, L.; Garg, A.; Wu, C.; Khadkevich, M.; Oswald, N.; Xie, E.; Egert, D.; Sreenivas, S. T.; Diao, S.; Yu, C.; Yu, Y.; Chen, W.; Norouzi, S.; Liu, J.; Lan, S.; Zhu, L.; Wang, J.; Jiang, J.; Mardani, M.; Maghoumi, M.; Han, S.; Jukić, A.; Tajbakhsh, N.; Kautz, J.; and Molchanov, P. 2026a. Nemotron-Labs-Diffusion: A Tri-Mode Language Model Unifying Autoregressive, Diffusion, and Self-Speculation Decoding. Technical report, NVIDIA. Technical report.

Fu, Y.; Whalen, L.; Ye, Z.; Dong, X.; Diao, S.; Liu, J.; Wu, C.; Zhang, H.; Xie, E.; Han, S.; Khadkevich, M.; Kautz, J.; Lin, Y. C.; and Molchanov, P. 2026b. Efficient-DLM: From Autoregressive to Diffusion Language Models, and Beyond in Speed. arXiv:2512.14067.

Ghazvininejad, M.; Levy, O.; Liu, Y.; and Zettlemoyer, L. 2019. Mask-Predict: Parallel Decoding of Conditional Masked Language Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Gong, S.; Li, M.; Feng, J.; Wu, Z.; and Kong, L. 2023. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Gu, J.; Bradbury, J.; Xiong, C.; Li, V. O. K.; and Socher, R. 2018. Non-Autoregressive Neural Machine Translation. In *International Conference on Learning Representations (ICLR)*.

Gulrajani, I.; and Hashimoto, T. B. 2023. Likelihood-Based Diffusion Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Jain, N.; Han, K.; Gu, A.; Li, W.-D.; Yan, F.; Zhang, T.; Wang, S.; Solar-Lezama, A.; Sen, K.; and Stoica, I. 2024. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. *arXiv preprint arXiv:2403.07974*.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J. E.; Zhang, H.; and Stoica, I. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*.

Lee, J.; Mansimov, E.; and Cho, K. 2018. Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Leviathan, Y.; Kalman, M.; and Matias, Y. 2023. Fast Inference from Transformers via Speculative Decoding. In *International Conference on Machine Learning (ICML)*.

Li, S.; Zhu, Y.; Wang, C.; Yin, F.; Shi, S.; Wang, Y.; Zhang, Y.; Huang, Y.; Zheng, H.; and Zhang, Y. 2025a. SpecForge: Train speculative decoding models effortlessly and port them smoothly to SGLang serving. https://github.com/sgl-project/SpecForge.

Li, X. L.; Thickstun, J.; Gulrajani, I.; Liang, P.; and Hashimoto, T. B. 2022. Diffusion-LM Improves Controllable Text Generation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Li, Y.; Wei, F.; Zhang, C.; and Zhang, H. 2024a. EAGLE-2: Faster Inference of Language Models with Dynamic Draft Trees. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Li, Y.; Wei, F.; Zhang, C.; and Zhang, H. 2024b. EAGLE: Speculative Sampling Requires Rethinking Feature Uncertainty. In *International Conference on Machine Learning (ICML)*.

Li, Y.; Wei, F.; Zhang, C.; and Zhang, H. 2025b. EAGLE-3: Scaling up Inference Acceleration of Large Language Models via Training-Time Test. *arXiv preprint arXiv:2503.01840*.

Lin, J.; Tang, J.; Tang, H.; Yang, S.; Chen, W.-M.; Wang, W.-C.; Xiao, G.; Dang, X.; Gan, C.; and Han, S. 2024. AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*.

Liu, F.; Tang, Y.; Liu, Z.; Ni, Y.; Han, K.; and Wang, Y. 2024a. Kangaroo: Lossless Self-Speculative Decoding via Double Early Exiting.

Liu, J.; Dong, X.; Ye, Z.; Mehta, R.; Fu, Y.; Singh, V.; Kautz, J.; Zhang, C.; and Molchanov, P. 2025. TiDAR: Think in Diffusion, Talk in Autoregression. arXiv:2511.08923.

Liu, X.; Hu, L.; Bailis, P.; Stoica, I.; Deng, Z.; Cheung, A.; and Zhang, H. 2024b. Online Speculative Decoding. In *International Conference on Machine Learning (ICML)*.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101.

Lou, A.; Meng, C.; and Ermon, S. 2024. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution. In *International Conference on Machine Learning (ICML)*.

Meta AI. 2024. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/.

Miao, X.; Oliaro, G.; Zhang, Z.; Cheng, X.; Wang, Z.; Zhang, Z.; Wong, R. Y. Y.; Zhu, A.; Yang, L.; Shi, X.; Shi, C.; Chen, Z.; Arfeen, D.; Abhyankar, R.; and Jia, Z. 2024. 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 (ASPLOS)*.

Nathawani, D.; and NVIDIA. 2025. Nemotron Post-Training Dataset V2. https://huggingface.co/datasets/nvidia/Nemotron-Post-Training-Dataset-v2.

Nie, S.; Zhu, F.; You, Z.; Zhang, X.; Ou, J.; Hu, J.; Zhou, J.; Lin, Y.; Wen, J.-R.; and Li, C. 2025. Large Language Diffusion Models. *arXiv preprint arXiv:2502.09992*.

NVIDIA Corporation. 2022. NVIDIA H100 Tensor Core GPU Architecture. https://resources.nvidia.com/en-us-tensor-core/gtc22-whitepaper-hopper.

OpenAI. 2024. OpenAI o1 System Card. *arXiv preprint arXiv:2412.16720*.

Pope, R.; Douglas, S.; Chowdhery, A.; Devlin, J.; Bradbury, J.; Heek, J.; Xiao, K.; Agrawal, S.; and Dean, J. 2023. Efficiently Scaling Transformer Inference.

Project Numina (AI-MO). 2024. AIMO Validation AIME: 90 American Invitational Mathematics Examination problems from AIME 2022–2024. https://huggingface.co/datasets/AI-MO/aimo-validation-aime.

Ringel, L.; and Romano, Y. 2026. Accelerating Speculative Decoding with Block Diffusion Draft Trees. *arXiv preprint arXiv:2604.12989*.

Rozière, B.; Gehring, J.; Gloeckle, F.; Sootla, S.; Gat, I.; Tan, X. E.; Adi, Y.; Liu, J.; Remez, T.; Rapin, J.; et al. 2023. Code Llama: Open Foundation Models for Code. *arXiv preprint arXiv:2308.12950*.

Samarin, A.; Krutikov, S.; Shevtsov, A.; Skvortsov, S.; Fisin, F.; and Golubev, A. 2026. LK Losses: Direct Acceptance Rate Optimization for Speculative Decoding. In *International Conference on Machine Learning (ICML)*.

Sandler, J.; Christopher, J. K.; Hartvigsen, T.; and Fioretto, F. 2025. SpecDiff-2: Scaling Diffusion Drafter Alignment for Faster Speculative Decoding. *arXiv preprint arXiv:2511.00606*.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Savinov, N.; Chung, J.; Binkowski, M.; Elsen, E.; and van den Oord, A. 2022. Step-unrolled Denoising Autoencoders for Text Generation. In *International Conference on Learning Representations (ICLR)*.

Stern, M.; Shazeer, N.; and Uszkoreit, J. 2018. Blockwise Parallel Decoding for Deep Autoregressive Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Sun, M.; Liu, Z.; Bair, A.; and Kolter, J. Z. 2024. A Simple and Effective Pruning Approach for Large Language Models. In *International Conference on Learning Representations (ICLR)*.

Wu, T.; Yao, Y.; Qi, Z.; Zheng, H.; Wang, Z.; Ma, H.; Liao, L.; Lakkaraju, H.; Li, J.; and Du, Y. 2026. D-PACE: Dynamic Position-Aware Cross-Entropy for Parallel Speculative Drafting. *arXiv preprint arXiv:2605.18810*.

Xiao, G.; Lin, J.; Seznec, M.; Wu, H.; Demouth, J.; and Han, S. 2023. SmoothQuant: Accurate and Efficient Post-Training

Quantization for Large Language Models. In *International Conference on Machine Learning (ICML)*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*.

Yu, G.-I.; Jeong, J. S.; Kim, G.-W.; Kim, S.; and Chun, B.-G. 2022. Orca: A Distributed Serving System for Transformer-Based Generative Models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*.

Zhang, J.; Wang, J.; Li, H.; Shou, L.; Chen, K.; Chen, G.; and Mehrotra, S. 2024a. Draft & Verify: Lossless Large Language Model Acceleration via Self-Speculative Decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhang, L.; Wang, X.; Huang, Y.; and Xu, R. 2024b. Learning Harmonized Representations for Speculative Sampling. *arXiv preprint arXiv:2408.15766*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*.

Zhou, Y.; Lyu, K.; Rawat, A. S.; Menon, A. K.; Rostamizadeh, A.; Kumar, S.; Kagénäck, J.-F.; and Agarwal, R. 2024. DistillSpec: Improving Speculative Decoding via Knowledge Distillation. In *International Conference on Learning Representations (ICLR)*.