SPECULATIVE DIFFUSION IS A TALE OF TWO TRADEOFFS

Anonymous authors
Paper under double-blind review

ABSTRACT

Speculative decoding is a technique that accelerates the inference speed of autoregressive large language models (LLMs) by sampling from a faster, lower quality draft model. Traditionally, a fundamental characteristic of speculative decoding is that it is lossless with respect to the verifier distribution, guaranteeing speed without sacrificing quality. This differs from recently developed diffusion LLMs (dLLMs), which are notably able to achieve substantial speedups by degrading the quality of their outputs with less diffusion effort. Because diffusion models are able to sample faster than autoregressive models by generating tokens in parallel, they are considered strong candidates for draft models in speculative diffusion. While it is widely understood that diffusion models can sacrifice quality for speed, speculative decoding itself can also be generalized to generate lossy samples faster using more lenient acceptance criteria with respect to the verifier distribution. Presented with the two parameters of diffusion effort and speculative decoding leniency, this paper explores the full achievable Pareto frontier via joint variations of these parameters. Our experiments show that jointly varying diffusion effort and acceptance leniency can extend the speed–quality Pareto frontier over single step diffusion drafting.

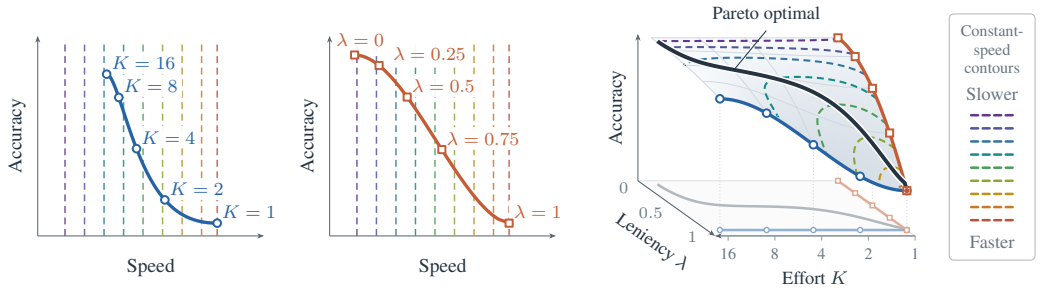
1 INTRODUCTION

Autoregressive (AR) large language models (LLMs) generate its output token by token, as the distribution of the next token over the vocabulary depends on all the previously generated tokens (Vaswani et al., 2017). Because each sequential decoding step requires transferring all model parameters and the growing key-value (KV) cache for a single token, inference is severely memory-bandwidth bound (Yuan et al., 2024). Thus, sequential dependency is a major bottleneck on the inference speed of AR decoding (Stern et al., 2018; Leviathan et al., 2023).

In light of this problem, speculative decoding was invented to speed up the inference of AR LLMs while preserving the original output distribution (Leviathan et al., 2023; Chen et al., 2023). It employs a much faster model as a drafter (Xia et al., 2024) to generate a short candidate sequence during each drafting round. The original LLM acts as a verifier and processes these proposed tokens in parallel via rejection sampling, accepting an initial prefix while strictly guaranteeing that the sampled distribution matches that of the verifier (Leviathan et al., 2023; Chen et al., 2023). It is known that the rejection probability at a given position is the total variation (TV) distance between the drafter distribution and the verifier distribution at that position (Leviathan et al., 2023; Chen et al., 2023; Yin et al., 2024). Therefore, a drafter more aligned with the verifier will have a higher expected acceptance rate across the sequence and a longer accepted prefix in each drafting round. Because only one verifier call is made during each drafting round, a longer accept length will lead to a higher throughput in output tokens per unit time. Because the resulting sequence is provably drawn from the verifier model's output distribution, traditional speculative decoding is *lossless*, providing an exact distributional guarantee that yields acceleration without degrading generation quality.

While this exact distributional guarantee ensures that model quality is fully preserved, lossless decoding limits the maximum achievable speedup by how closely the drafter can approximate the verifier. Various methods explore *lossy* speculative decoding, deliberately relaxing the exact distributional guarantee to trade off output quality for faster inference. For example, Leviathan et al. (2023) experimented with a lenience parameter that increases the acceptance rate of draft tokens by

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.



(a) Diffusion **(b) Speculative decoding** **(c) Speculative diffusion**

Figure 1: **A conceptual overview.** Diffusion has a speed-quality tradeoff parameterized by diffusion effort K, while speculative decoding has a speed-quality tradeoff parameterized by acceptance leniency λ. This work argues that speculative diffusion exposes both tradeoffs and therefore reveals optimal configurations that could not be found by varying only a single parameter.

a predefined ratio. Cactus (Hao & Mou, 2026) formalizes a relaxation by constraining the distributional divergence (such as Kullback-Leibler divergence) between the drafter and verifier to remain beneath a specified threshold. These lossy formulations provide a mechanism to control the trade-off between output quality and speed.

In parallel with developments in speculative decoding, diffusion large language models (dLLMs) have emerged as an alternative to autoregressive models. Rather than decoding from left to right, dLLMs iteratively denoise a noise corrupted sequence over multiple steps. Diffusion language models can be categorized by their noising process into masked diffusion (Sahoo et al., 2024; Nie et al., 2025; Ye et al., 2025) and uniform diffusion (Austin et al., 2021; Sahoo et al., 2025; DiffusionGemma Team, 2026). By updating all token positions simultaneously during each denoising step, diffusion models avoid the sequentiality of autoregressive decoding and instead decode tokens in parallel. Crucially, diffusion language models possess an intrinsic speed-quality trade-off: decreasing the number of denoising steps increases speed at the cost of quality.

More recently, an increasing number of diffusion LLMs have been used in speculative decoding. Due to their ability to generate multiple tokens simultaneously, diffusion LLMs serve as natural candidates as draft models. We refer to speculative decoding with a diffusion drafter as *speculative diffusion* (Christopher et al., 2025; Liu et al., 2026; Chen et al., 2026; Fu et al., 2026). Speculative diffusion presents two intertwined trade-offs: the drafter trades draft quality for speed via the number of denoising steps, while the verifier can trade quality for speed via lossy acceptance. Mapping out this joint optimization space reveals how both parameters interact to yield the final Pareto frontier between generation speed and output quality (Figure 1).

To address this challenge, this paper makes the following contributions:

(i) **Theoretical.** We prove that, for a fixed proposal distribution, linear interpolation between the verifier and draft distributions maximizes token acceptance under a total variation bound on deviation from the verifier (Theorem 1). We also establish conditions under which rejection sampling preserves the verifier distribution for multistep diffusion drafts (Proposition 2).

(ii) **Empirical.** Across masked and uniform diffusion drafters on reasoning and coding tasks, we show that different combinations of denoising effort and acceptance leniency contribute to the observed speed-accuracy Pareto frontier. In particular, additional denoising can support more lenient acceptance and faster generation at comparable accuracy (Section 4). We show that joint variation over these parameters often extends the Pareto frontier of speculative diffusion.

2 BACKGROUND

The contributions of this work depend on diffusion language models and lossy speculative decoding. We first provide background on these methods.

Diffusion Language Models. Diffusion language models iteratively generate text from noise. For a token vocabulary $\mathcal{V}$ and sequence of B tokens $\mathbf{x} \in \mathcal{V}^B$, let $\mathbf{z}_t$ be its corrupted version at noise level $t \in [0, 1]$. The forward noising process independently sets each z_t^i to a noisy token with probability t. Masked diffusion sets a special mask symbol $z_t^i = [\texttt{MASK}]$ (Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024), while uniform diffusion samples $z_t^i \sim \mathrm{Uniform}(\mathcal{V})$ (Sahoo et al., 2025; Von Rütte et al., 2025). The goal of diffusion language models is to learn a denoiser f_θ that can reverse the forward noise process. In this work, we generate one block of B tokens at a time (Arriola et al., 2025), conditioned on a fixed prefix h. The denoiser $f_\theta(\mathbf{z}_t \mid h)$ outputs a probability distribution over $|\mathcal{V}|$ over clean tokens at each block position in parallel. To generate a block, a dLLM starts from noise and iteratively denoises the block until termination. We refer to the number of diffusion steps K as *diffusion effort*. Similar to "reasoning effort" in reasoning models (Guo et al., 2025), greater effort applies more computation to generate the text, typically improving quality at the cost of inference speed.

Lossless Speculative Decoding. Speculative decoding is a rejection sampling method to draw samples from a distribution p using a proposal distribution q. Let $p(v \mid h)$ and $q(v \mid h)$ denote the verifier and draft distributions over a vocabulary $\mathcal{V}$, given a prefix h. Speculative decoding accepts a proposal $y \sim q(\cdot \mid h)$ with probability $\min\{1, p(y \mid h)/q(y \mid h)\}$, and upon rejection, samples a replacement with probability proportional to $[p(v \mid h) - q(v \mid h)]_+$, where $[a]_+ = \max\{a, 0\}$. This rule preserves the verifier distribution (Leviathan et al., 2023; Chen et al., 2023). Let $\alpha(h)$ denote the probability of accepting a proposal given accepted prefix h. Under lossless verification,

$$\alpha(h) = \sum_{v \in \mathcal{V}} \min\{p(v \mid h), q(v \mid h)\} = 1 - D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h)). \tag{1}$$

where D_{TV} defines the total variation distance between p and q.

$$D_{\mathrm{TV}}(p, q) = \frac{1}{2} \sum_{v \in \mathcal{V}} |p(v) - q(v)|. \tag{2}$$

Thus, the acceptance rate of lossless speculative decoding is predetermined by the alignment of the drafter q and verifier p. In practice, the space of possible prefixes is too large to estimate $\alpha(h)$ for every prefix h. We therefore measure the expected acceptance rate over a distribution of prefixes $\mathcal{H}$.

$$\bar{\alpha} = \mathbb{E}_{h \sim \mathcal{H}}[\alpha(h)], \tag{3}$$

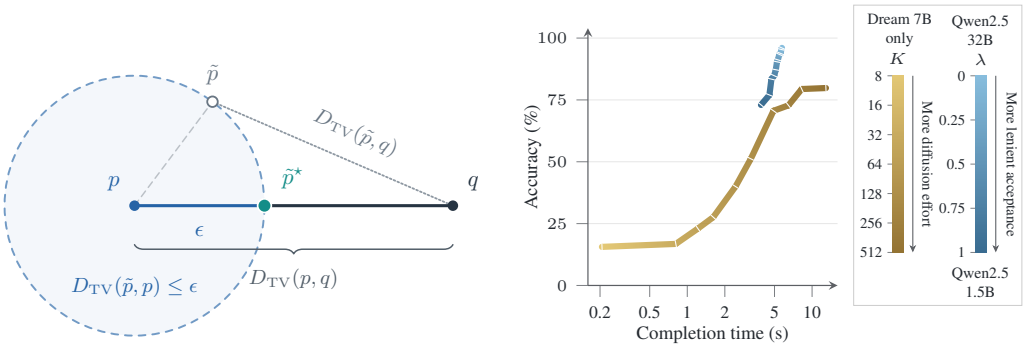
which can be estimated with sampling.

Lossy Speculative Decoding. Lossy speculative decoding relaxes the requirement of sampling exactly from p to increase acceptance rate. The degree of relaxation is the *acceptance leniency*, as introduced by Leviathan et al. (2023, Appendix A.5). In particular, we consider linear interpolation between the verifier and draft distributions (Fu et al., 2025; Hao & Mou, 2026; Wang et al., 2026) and characterize its optimality under a total variation constraint.

Suppose the target p can be replaced by a modified target $\tilde{p}$ in the rejection-sampling rule of Section 2. If $\tilde{p}(\cdot \mid h)$ is chosen before drawing $y \sim q(\cdot \mid h)$, rejection sampling can be performed using $\tilde{p}(\cdot \mid h)$ with an acceptance rate of $\alpha(h) = 1 - D_{\mathrm{TV}}(\tilde{p}(\cdot \mid h), q(\cdot \mid h))$. Observe that the distance between $\tilde{p}$ and q determines acceptance, while the distance between $\tilde{p}$ and p corresponds to the deviation from the original output distribution p. Given a bound $\epsilon \geq 0$, the best achievable acceptance is

$$\max_{\tilde{p}(\cdot|h) \in \Delta(\mathcal{V})} 1 - D_{\mathrm{TV}}(\tilde{p}(\cdot \mid h),\, q(\cdot \mid h)) \quad \text{subject to} \quad D_{\mathrm{TV}}(\tilde{p}(\cdot \mid h),\, p(\cdot \mid h)) \leq \epsilon \tag{4}$$

where $\Delta(\mathcal{V})$ is the set of probability distributions on the vocabulary.



(a) Optimal relaxed target distribution

(b) Motivating example

Figure 2: **Illustrative examples.** Left: Visualization of the proof of Theorem 1 shows that the linear interpolated distribution $\tilde{p}^\star$ is the optimal solution to maximize drafter q and verifier p alignment while staying within a constraint on deviation from the verifier ϵ. The solution to this constrained optimization is $\tilde{p}$. Right: A motivating example shows that the speed-quality frontiers independently induced by diffusion effort and acceptance lenience in speculative decoding can occupy separate regions of the Pareto frontier. We compare diffusion only with AR speculative decoding with leniency.

Theorem 1 (Optimality of Linear Acceptance)**.** For a fixed prefix h with $p(\cdot \mid h) \neq q(\cdot \mid h)$ and a TV budget $0 \leq \epsilon \leq D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h))$, an optimal target in Equation 4 is

$$\tilde{p}^\star(v \mid h) = (1-\lambda)p(v \mid h) + \lambda q(v \mid h), \qquad \lambda = \frac{\epsilon}{D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h))} \tag{5}$$

and achieves maximum acceptance probability $\alpha^\star(h) = 1 - D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h)) + \epsilon$.

In the above theorem, λ dictates the acceptance leniency by shifting the target toward the drafter. Lossless sampling from the verifier occurs at $\lambda = 0$, while $\lambda = 1$ samples from the draft distribution. Figure 2a gives an intuitive visual proof of the theorem. Total variation distance is a distance metric, so geometrically the optimal path between two distributions will be a linear interpolation. The full proof is given in Appendix A.

Note that this result characterizes the best acceptance obtainable by choosing a target distribution *before* observing the proposed token. Another point of clarification in the above proof is that it is a statement on the next token distribution, which assumes conditioning on prefix h. These bounds are not trivially extendable to the joint distribution of p and q over multiple token positions.

Cactus (Hao & Mou, 2026) is another example that allows lossy acceptance. Unlike linear acceptance, Cactus controls leniency by modifying the verifier distribution *after* observing a proposed token. It increases the probability of that token subject to a bound on the modified target's KL divergence, or any other f-divergence, from the verifier, with larger budgets permitting more lenient acceptance. This dependence on the sampled token distinguishes Cactus from the linear acceptance defined above, whose target is chosen before sampling.

3 Speculative Diffusion

Because diffusion models can draft tokens in parallel, speculative diffusion is a promising approach to achieve fast and high quality LLM inference. As discussed previously, both diffusion and speculative decoding have inherent speed quality tradeoffs.

3.1 A Motivating Example

To motivate varying diffusion denoising effort and speculative decoding leniency jointly, it is useful to examine each parameter independently. This work is motivated by the following observations.

Algorithm 1 Speculative diffusion

Require: prompt h; block size B; denoising steps K;
denoiser f_θ; verifier p; target rule RELAXTARGET

while EOS $\notin h$ **do**
  $(y_{1:B}, q_{1:B}) \leftarrow$ DIFFUSIONDRAFT(f_θ, h, B, K)
  $p_{1:B+1} \leftarrow$ VERIFIERFORWARD$(p, h, y_{1:B})$
  $\tilde{p}_{1:B} \leftarrow$ RELAXTARGET$(p_{1:B}, q_{1:B}, y_{1:B})$
  $k \leftarrow 0$
  do
    $k \leftarrow k + 1$
    $u \sim \mathcal{U}(0, 1)$
  while $k \leq B$ **and** $u \leq \tilde{p}_k(y_k)/q_k(y_k)$
  $y_k \sim \begin{cases} \text{NORMALIZE}([\tilde{p}_k - q_k]_+), & k \leq B, \\ p_k, & k = B + 1. \end{cases}$
  $h \leftarrow$ APPEND$(h, y_{1:k})$
end while
return TRUNCATE(h)

First, even with a large leniency under lossy speculative decoding and a very small AR drafter, the range of achievable drafting speeds is much narrower than that of in a dLLM when varying the diffusion effort. For example, Figure 2b shows that ignoring the verifier and completely relying on the AR drafter distribution Qwen2.5-1.5B-Instruct (Qwen Team, 2024a) achieves a speedup, but does cannot capture the full frontier. Meanwhile, a diffusion model is capable of can running over $10\times$ faster. This suggests dLLM drafters expose a new axis of variation on its drafting quality and speed in a way that traditional AR drafters cannot.

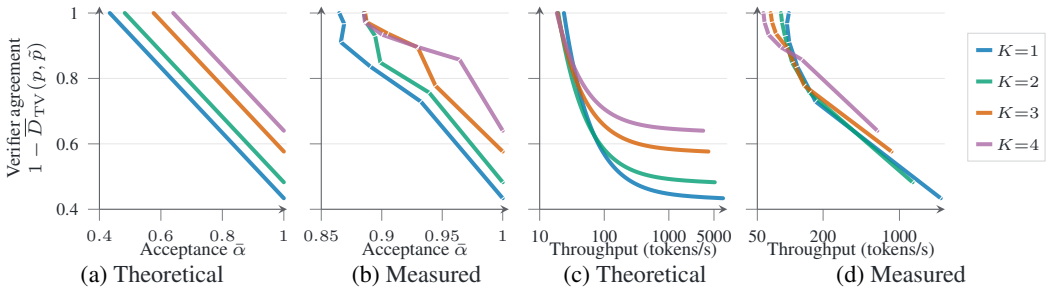
Second, even though varying diffusion effort with the dLLM alone can yield a speed and quality on the Pareto frontier, most dLLMs have significantly lower accuracies than those of the state-of-the-art AR LLMs. For example, in Figure 2b the accuracy of Dream 7B plateaus despite running a high number of diffusion steps. Therefore, a way to uplift the accuracy of the diffusion LLMs with an AR model as a guide would be desirable. Many recent works echo this sentiment (Liu et al., 2025; Israel et al., 2025; Hu et al., 2026).

With these observations, it is evident that the speed quality tradeoffs of diffusion and lossy speculative decoding are complementary. Diffusion alone can achieve higher speeds. Speculative decoding alone can achieve higher quality. Because each method is Pareto optimal independently, it is possible that combining the approaches can unlock further improvements in the Pareto frontier of speed and quality.

3.2 MULTI-STEP DIFFUSION DRAFTING

Many instances of speculative diffusion focus on single step diffusion drafting Fu et al. (2026); Sahoo et al. (2026); Chen et al. (2026); Cheng et al. (2026b). In this work, we use a diffusion drafter for multiple steps per verifier call, as shown in Algorithm 1. While diffusion effort can be parameterized directly in the number of steps K (as mentioned in Section 2), other criteria can also be used as parametrization to control the diffusion effort. For example, confidence thresholds (Nie et al., 2026) or entropy bounds (Ben-Hamu et al., 2026) can indirectly affect the number of diffusion steps.

In speculative decoding with an autoregressive drafter, when sampling token at position i it is conditioned on tokens prior to position i that have been accepted by the verifier. Multi-step diffusion drafting does not maintain this property. A final sampled token can be obtained through many possible denoising trajectories. Evaluating the marginal probability at a position would require summing the probability over an intractable number of trajectories or approximating via an evidence lower bound (ELBO) (Zhao et al., 2025). Therefore, it is not immediately clear what proposal probabilities should be used in rejection sampling.



(a) Theoretical (b) Measured (c) Theoretical (d) Measured

Figure 3: **A model of speculative diffusion.** Theoretical and measured tradeoffs for Dream-7B drafting for Qwen2.5-32B on GSM8K. (a) The theoretical acceptance rate (speed) and verifier agreement (quality) under linear acceptance for each of K diffusion steps per round. (b) Empirically measured acceptance rate for each K. (c) A transformation from acceptance rate to throughput according to a simple model of throughput. (d) Empirically measured throughput.

Since the diffusion trajectory is stochastic, the proposal distribution itself can be seen as a random vector over the probability simplex $\Delta(\mathcal{V})$. Proposition 2 shows that without knowing the underlying distribution of the proposal distribution random vector, rejection sampling can still be performed on the proposal distribution that generated each token, maintaining exact samples from the verifier. In the context of speculative diffusion, this permits use of the final step probabilities (e.g. final transition probabilities in uniform diffusion or marginals in masked diffusion) as the proposal probabilities in rejection sampling.

3.3 A Simple Throughput Model

A theoretical model can aid our understanding in how speed and generation quality vary as a function of the diffusion effort and leniency. To this end, the expected acceptance rate $\bar{\alpha} = 1 - D_{\mathrm{TV}}(\tilde{p}, q)$ can be used to model throughput and the similarity between the verifier distribution and the modified target distribution $1 - D_{\mathrm{TV}}(\tilde{p}, p)$ can be used as a proxy for generation quality, where both distances can be derived from an empirically profiled $D_{\mathrm{TV}}(p, q)$ using Theorem 1 at different effort levels. For diffusion effort using K steps per drafting round, let $\mathcal{H}_K$ be a fixed distribution of prefixes and q_K the corresponding draft distribution. Averaging over the distribution of prefixes, we define $d_K = \mathbb{E}_{h \sim \mathcal{H}_K}[D_{\mathrm{TV}}(p(\cdot \mid h), q_K(\cdot \mid h))]$, which is the TV distance between the verifier and drafter at effort K with zero leniency. Applying the linear acceptance rule in Theorem 1 over H_K gives the expected acceptance rate and similarity as

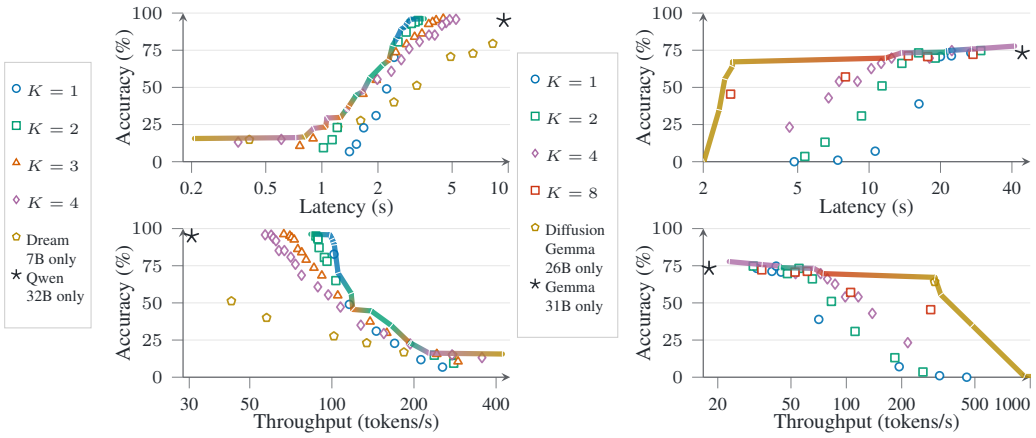
$$\bar{\alpha}_K(\lambda) = 1 - (1 - \lambda) d_K \qquad \text{and} \qquad 1 - \overline{D}_{\mathrm{TV}}(p, \tilde{p}) = 1 - \lambda d_K. \tag{6}$$

Let $c_d(K)$ and c_v denote the time it takes to call the drafter and verifier per drafting round respectively. Empirically, $c_d(K) \approx a + bK$ and c_v is constant. Proposition 1 in Appendix B then gives the theoretical throughput as

$$\mathcal{T}(K, \bar{\alpha}) = \frac{\sum_{j=0}^{B} \bar{\alpha}^j}{c_d(K) + c_v}. \tag{7}$$

Figures 3b and 3d directly measure each quantity predicted in Figures 3a and 3c. The theoretical model captures two important qualitative features seen in the empirically measured results: (1) The tradeoff between acceptance rate and verifier agreement is approximately linear (2) no denoising budget captures the full throughput-quality Pareto frontier. The quantitative discrepancy between theoretical and measured results is primarily driven by the incorrect assumption of a universal acceptance rate $\bar{\alpha}$. Leniency changes which draft tokens are accepted and hence the prefixes on which subsequent predictions are conditioned. More formally, the prefix distribution $\mathcal{H}_K$ encountered during the empirical profiling differs from the prefix distribution at higher leniency, resulting in a different value of d_K. Another difference between theory and practice is that the theory computes acceptance over an entire block, but in practice throughput can be affected by EOS termination effectively reducing drafting block size.

This simple setup illustrates the qualitative features of speculative diffusion when varying diffusion effort and acceptance leniency. However, it is not obvious how quality measured by total variation



(a) GSM8K — Dream–Qwen (b) GPQA — DiffusionGemma–Gemma

Figure 4: **Examples of the Pareto frontier.** We provide examples of speculative diffusion for both masked diffusion (left) and uniform diffusion (right). Each point signifies a configuration of acceptance leniency in verification and diffusion effort (K steps) in drafting. We observe the same qualitative trend that the speed-quality Pareto frontier is occupied by many different settings of diffusion effort which require different levels of lenience.

translates to downstream metrics such as reasoning accuracy. Furthermore, the linear interpolation acceptance criterion enables simpler modeling of acceptance rate, but other acceptance criteria may not behave the same way. These questions will be addressed in the following experimental section.

4 Experiments

The goal of our experiments is to empirically explore the two tradeoffs of speculative diffusion: diffusion effort and leniency. We first establish a common experimental setup, then conduct experiments in two different settings.

4.1 Experimental Setup

We first examine masked and uniform diffusion using the following model pairs: Dream-7B (Ye et al., 2025) with Qwen2.5-32B and DiffusionGemma-26B-A4B (DiffusionGemma Team, 2026) with Gemma4-31B (Gemma Team, 2026). We then compare several other recent speculative diffusion models under a common Qwen3-8B verifier.

In general, we benchmark quality using GSM8K (Cobbe et al., 2021), GPQA Diamond (Rein et al., 2024), and HumanEval (Chen et al., 2021) for math, reasoning, and code. In our Dream-Qwen and DiffusionGemma-Gemma experiments, some model-benchmark combinations are excluded based on two criteria that depend on the standalone capabilities of the drafter and verifier.

The first criterion is that if the standalone accuracies of the drafter and verifier models are nearly the same on a benchmark, then there is no benefit to running speculative decoding, and thus the combination will be excluded. The second criterion is that if the drafter model performs worse than random, then the combination will be excluded.

Table 1 reports the baseline accuracies for both the standalone drafter and verifier of the Dream, SDAR, and DiffusionGemma pairs, on GSM8K, HumanEval, and GPQA Diamond. Based on these measurements, we evaluate Dream and SDAR on GSM8K and HumanEval, and DiffusionGemma on GPQA. An evaluation configuration consists of a benchmark, model pair, sampling configuration, diffusion effort, and acceptance leniency. The detailed experimental configurations can be found in the Appendix E.

4.2 Joint Exploration Over Two Tradeoffs

Figure 4 is obtained by fixing a denoising budget K and varying the acceptance leniency. In this particular plot, we display the results using the modified Cactus acceptance criterion instead of linear acceptance to demonstrate the generality of our findings. The corresponding linear-acceptance results are in Figures 7b and 8b and are qualitatively similar. See Appendix D for details.

Every point in Figure 4 represents the speed and accuracy measured from a diffusion effort, acceptance leniency pair. We then interpolate the Pareto frontier over all pairs for latency and throughput. Note that latency is wall-clock time to problem completion, while throughput is the number of generated tokens divided by the wall clock time to problem completion. Both are needed to quantify speed, because throughput can be over optimized by generating unnecessary tokens, and we do not want to achieve latency improvements entirely from shorter generation lengths. The Pareto frontier is visualized as a thick connected line, while hollow markers are measurements not on the frontier. For visual clarity, we omit the points on the Pareto frontier itself and instead mark them with white ticks on the line. The Pareto frontier's color identifies the diffusion effort K in that region of the frontier. Between measurements of the same color, the line maintains that color, and between different colors we apply a color gradient. This makes it possible to see which denoising budgets contribute to the frontier.

For both model pairs in Figure 4, several values of K contribute to the frontier. Varying leniency at a single diffusion effort configuration will fail to achieve Pareto-optimal results. In general, low diffusion effort with strict acceptance requirements tends to capture the higher accuracy regions of the Pareto frontier, while higher diffusion effort with more lenient acceptance captures the higher speed regions. This is natural because as previously observed, diffusion has higher speed but lower quality and autoregression has higher quality but lower speed.

4.3 Comparing Drafters with a Shared Verifier

While the previous experimental setup analyzes speculative diffusion for model pairs that were trained separately, many recent works train specifically for diffusion speculation. Under a Qwen3 8B (Yang et al., 2025) verifier, these works include Uno (Sahoo et al., 2026), DFlash (Chen et al., 2026), and DSpark (Cheng et al., 2026b). For this experiment we also include SDAR (Cheng et al., 2026a), which notably was not trained for the task of speculative decoding.

We compare SDAR-1.7B, SDAR-8B, Uno, DFlash, and DSpark on GSM8K and HumanEval with a shared Qwen3-8B verifier. For each model, we apply linear acceptance at $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$, from lossless verification to full acceptance. Uno, DFlash, and DSpark are trained specifically for one step diffusion drafting. DFlash and DSpark apply feature conditioning from the verifier, making multiple diffusion steps per verifier call impossible. Thus, for these models we only include their Pareto frontier using one step drafting and acceptance lenience. For Uno, we evaluate $K \in \{1, 2, 4\}$ in Appendix F.4, but we find only $K = 1$ one-step diffusion occupies its Pareto frontier. Because SDAR is trained as a standalone diffusion model with a multistep objective, we measure it's Pareto frontier by varying diffusion steps as well. We vary SDAR's adaptive denoising effort using a confidence threshold $\tau \in \{0.3, 0.5, 0.7, 0.9\}$ and also evaluate fixed one-step ($K = 1$) drafts. For each model size, we compare the joint frontier over all these settings with the frontier restricted to $K = 1$. Appendix F.3 shows all measured configurations. The results are shown in Figure 5.

Joint variation over diffusion effort and acceptance leniency substantially improves SDAR's speed-accuracy tradeoff. While the specialized one-step drafters remain strong, especially in throughput, joint variation places SDAR on the frontier despite not having specialized training and architecture for speculative diffusion. These results motivate training more specialized drafters for multi-step refinement.

5 Related Work

Diffusion drafters. Recent work on diffusion drafting for speculative decoding aims to generate proposals with fewer sequential model evaluations. SpecDiff introduced diffusion drafters and studied how denoising steps balance drafting cost against acceptance (Christopher et al., 2025). Subse-

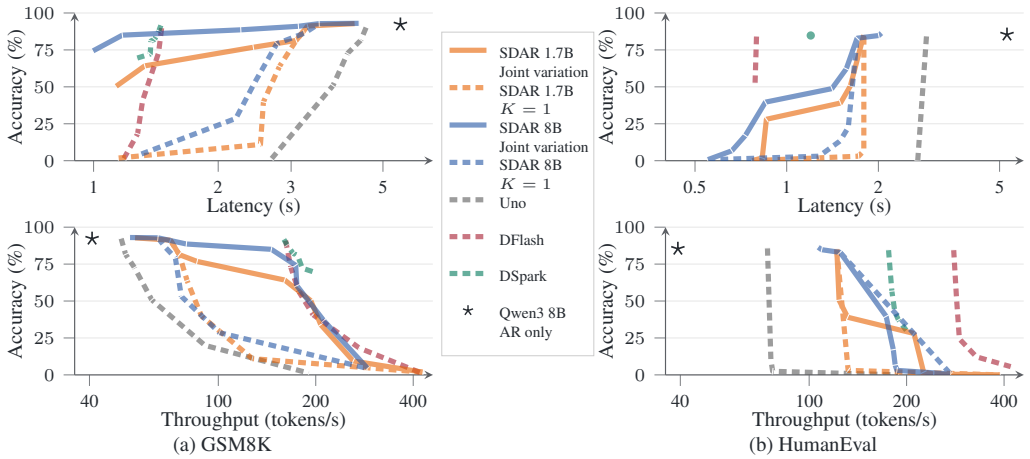


Figure 5: **Comparison of Pareto frontiers.** For a single Qwen3 8B verifier, we benchmark several speculative diffusion methods. Each line corresponds to the achievable speed-quality frontier of a specific draft model. Dashed curves fix one diffusion step and vary leniency, while solid curves are varied over diffusion effort and acceptance lenience. For the SDAR models that support multi-step speculative diffusion, varying over the parameters jointly significantly extends the Pareto frontier.

quent work improves draft-verifier alignment through causal-consistency path search (Li et al., 2025) or conditioning diffusion drafts on target-model features (Chen et al., 2026), while hybrid models integrate diffusion drafting and autoregressive verification within one model (Liu et al., 2026; Fu et al., 2026; Sahoo et al., 2026). In general, diffusion drafting in speculative decoding rarely applies multiple steps of diffusion effort. Quadratic decoding (Samragh et al., 2025) applies multiple steps of diffusion for the purpose of tree-based verification (Miao et al., 2024). Tree-based speculative decoding is orthogonal to acceptance leniency and is beyond the scope of this work.

Lossy speculative decoding. Multiple works have introduced leniency into speculative decoding acceptance. It was first introduced in the seminal speculative decoding paper by Leviathan et al. (2023). Later theoretical work formalized the tradeoff as a linear relationship between rejection rates and total variation loss (Yin et al., 2024). In this paper, we spotlight the linear acceptance criterion which has been mentioned in several previous works under different names: collaborative decoding (Fu et al., 2025), linear interpolation (Hao & Mou, 2026), and dynamic ensemble verification (Wang et al., 2026). While the method is not original, to the best of our knowledge, our formulation is the first analyze it from the perspective of constrained optimization and show that it is optimal under a total variation constraint. Furthermore, these works do not investigate leniency in the context of speculative diffusion. Nemotron-Labs-Diffusion briefly describes a verifier that interpolates AR and diffusion distributions (Fu et al., 2026, Appendix C), but it is mentioned as a possible direction without experimental evaluation.

6 Conclusion

This work demonstrates that speculative decoding with diffusion drafters, also known as speculative diffusion, is governed by two complementary mechanisms that trade speed for quality. Specifically, these mechanisms are the denoising effort of the drafter and the acceptance leniency of the verifier. Our cost model and empirical findings show that neither parameter independently can capture the full Pareto frontier between generation speed and task accuracy. The speed quality trade-offs that speculative diffusion expand the design space for large language model serving. A promising future direction is to improve query routing, where incoming requests are mapped to distinct configurations along the Pareto frontier according to their task difficulty and tolerance for approximation. Beyond routing at the query level, future systems could employ dynamic scheduling policies that adjust denoising budgets and verification thresholds at runtime across generation rounds.

AI USAGE DISCLOSURE

The authors used LLMs to assist in finding relevant papers, plotting results, polishing LaTeX, and drafting code. We take responsibility for the final content of this work.

REFERENCES

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, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas S Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *Advances in Neural Information Processing Systems*, 38:55981–56007, 2026.

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. In *Proceedings of the 43rd International Conference on Machine Learning*, 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.

Shuang Cheng, Yihan Bian, Dawei Liu, Yuhua Jiang, Yihao Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 22058–22075, 2026a.

Xin Cheng, Xingkai Yu, Chenze Shao, Jiashi Li, Yunfan Xiong, Yi Qian, Jiaqi Zhu, Shirong Ma, Xiaokang Zhang, Jiasheng Ye, et al. DSpark: Confidence-scheduled speculative decoding with semi-autoregressive generation. *arXiv preprint arXiv:2607.05147*, 2026b. URL https://arxiv.org/abs/2607.05147.

Jacob K Christopher, Brian R. Bartoldson, Tal Ben-Nun, Michael Cardei, Bhavya Kailkhura, and Ferdinando Fioretto. Speculative diffusion decoding: Accelerating language generation through diffusion. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 12042–12059, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.601. URL https://aclanthology.org/2025.naacl-long.601/.

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

DiffusionGemma Team. DiffusionGemma technical report. *arXiv preprint arXiv:2608.00146*, 2026.

Jiale Fu, Yuchu Jiang, Junkai Chen, Jiaming Fan, Xin Geng, and Xu Yang. Fast large language model collaborative decoding via speculation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 17764–17782. PMLR, 2025. URL https://proceedings.mlr.press/v267/fu25c.html.

Yonggan Fu, Lexington Whalen, Abhinav Garg, Chengyue Wu, Maksim Khadkevich, Nicolai Oswald, Enze Xie, Daniel Egert, Sharath Turuvekere Sreenivas, Shizhe Diao, Chenhan Yu, Ye Yu, Weijia Chen, Sajad Norouzi, Jingyu Liu, Shiyi Lan, Ligeng Zhu, Jin Wang, Jindong Jiang, Morteza Mardani, Mehran Maghoumi, Song Han, Ante Jukić, Nima Tajbakhsh, Jan Kautz, and Pavlo Molchanov. Nemotron-Labs-Diffusion: A tri-mode language model unifying autoregressive, diffusion, and self-speculation decoding. *arXiv preprint arXiv:2607.05722*, 2026.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yongchang Hao and Lili Mou. Cactus: Accelerating auto-regressive decoding with constrained acceptance speculative sampling. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2604.04987.

Zhanqiu Hu, Jian Meng, Yash Akhauri, Mohamed Abdelfattah, Jae-sun Seo, Zhiru Zhang, and Udit Gupta. Flashdlm: Accelerating diffusion language model inference via efficient kv caching and guided diffusion. In *International Conference on Learning Representations*, volume 2026, pp. 153529–153544, 2026.

Daniel Israel, Guy Van den Broeck, and Aditya Grover. Accelerating diffusion llms via adaptive parallel decoding. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 52870–52888. Curran Associates, Inc., 2025. doi: 10.52202/085713-1763. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/4c7d31b2da05a17f15101db5b37a1e24-Paper-Conference.pdf.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023.

Guanghao Li, Zhihui Fu, Min Fang, Qibin Zhao, Ming Tang, Chun Yuan, and Jun Wang. DiffuSpec: Unlocking diffusion language models for speculative decoding. *arXiv preprint arXiv:2510.02358*, 2025. URL https://arxiv.org/abs/2510.02358.

Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete copula diffusion. In *International Conference on Learning Representations*, volume 2025, pp. 88953–88979, 2025.

Jingyu Liu, Xin Dong, Zhifan Ye, Rishabh Mehta, Yonggan Fu, Vartika Singh, Ce Zhang, and Pavlo Molchanov. TiDAR: Think in diffusion, talk in autoregression. In *Proceedings of Machine Learning and Systems*, volume 8, 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. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

Qwen Team. Qwen2.5: A party of foundation models, September 2024b. URL https://qwenlm.github.io/blog/qwen2.5/.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

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.

Mohammad Samragh, Arnav Kundu, David Harrison, Kumari Nishu, Devang Naik, Minsik Cho, and Mehrdad Farajtabar. Your llm knows the future: Uncovering its multi-token prediction potential. *arXiv preprint arXiv:2507.11851*, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Dimitri Von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. *arXiv preprint arXiv:2503.04482*, 2025.

Ziyi Wang, Siva Rajesh Kasa, Ankith M S, Santhosh Kumar Kasa, Jiaru Zou, Sumit Negi, Ruqi Zhang, Nan Jiang, and Qifan Song. DIVERSED: Relaxed speculative decoding via dynamic ensemble verification. *arXiv preprint arXiv:2604.07622*, 2026. URL `https://arxiv.org/abs/2604.07622`.

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. In *Findings of the Association for Computational Linguistics: ACL 2024*, 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.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Ming Yin, Minshuo Chen, Kaixuan Huang, and Mengdi Wang. A theoretical perspective for speculative decoding algorithm. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Zhihang Yuan, Yuzhang Shang, Yang Zhou, Zhen Dong, Zhe Zhou, Chenhao Xue, Bingzhe Wu, Zhikai Li, Qingyi Gu, Yong Jae Lee, et al. Llm inference unveiled: Survey and roofline model insights. *arXiv preprint arXiv:2402.16363*, 2024.

Lingxiao Zhao, Xueying Ding, Lijun Yu, and Leman Akoglu. Unified discrete diffusion for categorical data. *Journal of Machine Learning Research*, 26(215):1–49, 2025.

A Proof of Theorem 1

Proof. For any feasible $\tilde{p}(\cdot \mid h)$, the triangle inequality gives

$$\begin{aligned} D_{\mathrm{TV}}(\tilde{p}(\cdot \mid h), q(\cdot \mid h)) &\geq D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h)) - D_{\mathrm{TV}}(\tilde{p}(\cdot \mid h), p(\cdot \mid h)) \\ &\geq D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h)) - \epsilon. \end{aligned}$$

Hence, the acceptance probability is at most

$$1 - D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h)) + \epsilon.$$

We must now show that this bound is attained. Consider

$$\tilde{p}^\star(\cdot \mid h) = (1-\lambda)p(\cdot \mid h) + \lambda q(\cdot \mid h), \qquad \lambda = \frac{\epsilon}{D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h))}.$$

Under the theorem's assumptions, $\lambda \in [0, 1]$. By the definition of total variation distance,

$$\begin{aligned} D_{\mathrm{TV}}(\tilde{p}^\star(\cdot \mid h), p(\cdot \mid h)) &= \frac{1}{2} \sum_{v \in \mathcal{V}} \big|\lambda\big(q(v \mid h) - p(v \mid h)\big)\big| \\ &= \lambda D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h)) = \epsilon, \end{aligned}$$

so $\tilde{p}^\star$ is feasible. Similarly,

$$\begin{aligned} D_{\mathrm{TV}}(\tilde{p}^\star(\cdot \mid h), q(\cdot \mid h)) &= \frac{1}{2} \sum_{v \in \mathcal{V}} \big|(1-\lambda)\big(p(v \mid h) - q(v \mid h)\big)\big| \\ &= (1-\lambda) D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h)) \\ &= D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h)) - \epsilon. \end{aligned}$$

Thus $\tilde{p}^\star$ attains the lower bound and therefore achieves acceptance

$$1 - D_{\mathrm{TV}}(p(\cdot \mid h), q(\cdot \mid h)) + \epsilon.$$

□

B Expected Number of Generated Tokens per Drafting Round

Proposition 1. Assume there is a draft sequence of length B. Let $\mathbb{1}_i$ be the indicator variable for whether token i is accepted conditioned on all previous tokens being accepted, where $i \in [B]$. Assume $\{\mathbb{1}_i\}$ are independent and let $\mathbb{P}(\mathbb{1}_i = 1) = P_i$, where P_i is another set of i.i.d. random variables. Let $\mu := \mathbb{E}P_i$. Then the expected number of generated tokens is

$$\sum_{n=0}^{B} \mu^n = \begin{cases} \dfrac{1-\mu^{B+1}}{1-\mu} & \text{for } \mu \neq 1, \\ 1 + B & \text{for } \mu = 1. \end{cases} \tag{8}$$

Proof. The expected number of accepted tokens for a draft sequence of length B is

$$A(B) = \mathbb{E}\big[\mathbb{1}_1 + \mathbb{1}_1\mathbb{1}_2 + \cdots + \mathbb{1}_1\mathbb{1}_2\cdots\mathbb{1}_B\big] = \sum_{n=1}^{B} \mathbb{E} \prod_{i=1}^{n} \mathbb{1}_i. \tag{9}$$

Since the indicator variables $\mathbb{1}_i$ are independent, we have

$$\mathbb{E}\Big[\prod_{i=1}^{n} \mathbb{1}_i \,\Big|\, P_1, \ldots, P_n\Big] = \mathbb{P}(\mathbb{1}_i = 1 \; \forall i \leq n) = \prod_{i=1}^{n} P_i, \tag{10}$$

and

$$\begin{aligned}
\mathbb{E}\Big[\prod_{i=1}^{n}\mathbb{1}_i\Big] &= \mathbb{E}\Big[\mathbb{E}\Big[\prod_{i=1}^{n}\mathbb{1}_i \,\Big|\, P_1,\ldots,P_n\Big]\Big] \quad \text{law of total expectation} \\
&= \mathbb{E}\prod_{i=1}^{n}P_i \quad \text{by Equation 10} \\
&= \prod_{i=1}^{n}\mathbb{E}P_i \quad \text{as } \{P_i\} \text{ are i.i.d.} \\
&= \mu^n \quad \text{as } \mu = \mathbb{E}P_i,
\end{aligned} \tag{11}$$

Therefore,

$$A(B) = \sum_{n=1}^{B}\mathbb{E}\prod_{i=1}^{n}\mathbb{1}_i = \sum_{n=1}^{B}\mu^n. \tag{12}$$

Each drafting round also commits one more token, which is either the correction or the bonus token, so the expected number of generated tokens per drafting round is

$$1 + A(B) = 1 + \sum_{n=1}^{B}\mu^n = \sum_{n=0}^{B}\mu^n. \tag{13}$$

□

C Rejection Sampling for Multi-Step Diffusion Drafting

Proposition 2. Let $Q : \Omega \to \Delta(\mathcal{V})$ be a random vector representing the final probability vector with which the draft token is sampled from. Q is of unknown distribution because the diffusion trajectory that produced the drafter probability vector is unknown.
Let $X : \Omega \to \mathcal{V}$ be a random variable that represents the outcome of performing rejection sampling with a fixed target distribution $\tilde{p}$ and the proposal distribution Q. Then

$$\mathbb{P}(X = v) = \tilde{p}(v) \quad \forall v \in \mathcal{V} \tag{14}$$

Proof of Proposition 2. Based on rejection sampling theory (Leviathan et al., 2023; Chen et al., 2023), conditioned on the actual probability Q with which the drafted token is sampled from, we know that

$$\mathbb{E}(\mathbb{1}_{X=v} \mid Q) = \mathbb{P}(X = v \mid Q) = \tilde{p}(v) \quad \forall v \in \mathcal{V}. \tag{15}$$

So by the law of total expectation, we have

$$\mathbb{P}(X = v) = \mathbb{E}(\mathbb{1}_{X=v}) = \mathbb{E}_q[\mathbb{E}(\mathbb{1}_{X=v} \mid Q = q)] = \mathbb{E}_q[\tilde{p}(v)] = \tilde{p}(v) \quad \forall v \in \mathcal{V}$$

Thus Equation 14 holds. □

D Modified Cactus

Given a prefix $\mathbf{x}_{<t}$ and a drafted token y with drafter probability $q(y \mid \mathbf{x}_{<t})$ and verifier probability $p(y \mid \mathbf{x}_{<t})$, Cactus (Hao & Mou, 2026) creates a modified target probability distribution $\tilde{p}$ that depends on y that maximizes $\tilde{p}(y \mid \mathbf{x}_{<t})$ subject to the constraint $D_{\mathrm{KL}}(\tilde{p}(\cdot \mid \mathbf{x}_{<t}) \,\|\, p(\cdot \mid \mathbf{x}_{<t})) \le \delta$. Specifically, it sets $\tilde{p}(y \mid \mathbf{x}_{<t}) = \gamma$, where γ is the solution to Equation 16, and rescales the assigned probabilities for the rest of the tokens so that the overall vector remains a probability distribution, i.e., $\tilde{p}(v \mid \mathbf{x}_{<t}) = \frac{1-\gamma}{1-p(y|\mathbf{x}_{<t})}\, p(v \mid \mathbf{x}_{<t})$ for $v \neq y$. For $0 < p(y \mid \mathbf{x}_{<t}) < 1$, reduces to

$$\Phi(\gamma) := \gamma \log \frac{\gamma}{p(y \mid \mathbf{x}_{<t})} + (1-\gamma)\log\frac{1-\gamma}{1-p(y \mid \mathbf{x}_{<t})} \le \delta. \tag{16}$$

Model pair	Benchmark	n	Drafter (%)	Verifier (%)	Gap (pp)
Dream 7B → Qwen2.5 32B	GSM8K	500	79.8 ± 1.8	95.0 ± 1.0	$+15.2$
	HumanEval	164	57.3 ± 3.9	84.8 ± 2.8	$+27.4$
	GPQA Diamond	198	22.7 ± 3.0	45.5 ± 3.5	$+22.7$
SDAR 8B → Qwen3 8B	GSM8K	500	85.0 ± 1.6	92.2 ± 1.2	$+7.2$
	HumanEval	164	39.6 ± 3.8	85.4 ± 2.8	$+45.7$
	GPQA Diamond	198	25.3 ± 3.1	43.4 ± 3.5	$+18.2$
DiffusionGemma 26B-A4B → Gemma-4 31B	GSM8K	500	96.4 ± 0.8	96.6 ± 0.8	$+0.2$
	HumanEval	164	91.5 ± 2.2	93.3 ± 2.0	$+1.8$
	GPQA Diamond	198	64.1 ± 3.4	73.2 ± 3.1	$+9.1$

Table 1: Baseline task accuracy with one standard error. The gap is verifier minus drafter accuracy in percentage points.

Cactus approximates Φ by its second-order Taylor expansion at $\gamma_0 = p(y \mid \mathbf{x}_{<t})$,

$$\Phi(\gamma) \approx \Phi(\gamma_0) + \Phi'(\gamma_0)(\gamma - \gamma_0) + \frac{\Phi''(\gamma_0)}{2}(\gamma - \gamma_0)^2, \tag{17}$$

and solves $\Phi(\gamma) = \delta$ under this approximation, which gives $\gamma = \min\{p(y \mid \mathbf{x}_{<t}) + \sqrt{2\delta p(y \mid \mathbf{x}_{<t})(1 - p(y \mid \mathbf{x}_{<t}))},\, 1\}$. Cactus then accepts y with probability $\min\{1, \gamma / q(y \mid \mathbf{x}_{<t})\}$.

However, the Taylor approximation is only justified when the desired modified probability for the drafted token is close to $\gamma_0 = p(y \mid \mathbf{x}_{<t})$.

In order to support larger leniency δ values, when the verifier probability $\gamma_0 = p(y \mid \mathbf{x}_{<t})$ is small and the drafter probability $q(y \mid \mathbf{x}_{<t})$ is large, the solution γ^* needs to be much larger than γ_0 to become comparable to the drafter probability, which will break the Taylor approximation.

Therefore, in our experiments we use Newton's method to solve for the optimal γ^* exactly for any given values of δ, $p(y \mid \mathbf{x}_{<t})$, and $q(y \mid \mathbf{x}_{<t})$. On the GPU, the Newton steps can be run in parallel for the entire drafting block, so the runtime overhead is minimal.

E Experimental Details

All experiments were run on a single H200 GPU.

E.1 Benchmark Selection

The benchmarks use a random set of 500 GSM8K questions, all GPQA Diamond and all HumanEval problems. GSM8K and GPQA score the extracted final answer, while HumanEval reports accuracy from execution of the generated code against its tests using pass@1. We generate exactly one completion for all problems.

Table 1 reports baseline accuracies used to select benchmarks. Dream uses 512 denoising steps on GSM8K and HumanEval and 1024 on GPQA. DiffusionGemma uses its native sampler with an upper limit of 48 diffusion steps. SDAR uses a confidence threshold of $\tau = 0.9$. These settings assess drafter quality with high diffusion effort. We focus on settings where the drafter has useful task accuracy and the verifier offers room for improvement. Dream and SDAR meet these criteria on GSM8K and HumanEval, but perform worse than random on GPQA. DiffusionGemma has almost the same accuracy as the verifier model than on GSM8K and Humaneval so it is only run on GPQA Diamond.

E.2 Models, Data, and Sampling

Table 2 shows the Huggingface checkpoints used for the experiments. All models, both drafters and verifiers, are run without thinking enabled. All models have a generation limit of 2048 for GPQA Diamond and 512 on GSM8K and HumanEval.

Model / role	Hugging Face checkpoint	Reference
Dream drafter	`Dream-org/Dream-v0-Instruct-7B`	Ye et al. (2025)
Qwen2.5 verifier	`Qwen/Qwen2.5-32B-Instruct`	Qwen Team (2024b)
DiffusionGemma drafter	`google/diffusiongemma-26B-A4B-it`	DiffusionGemma Team (2026)
Gemma verifier	`google/gemma-4-31B-it`	Gemma Team (2026)
Qwen3 verifier / Uno	`s-sahoo/uno-qwen3-8B`	Sahoo et al. (2026)
SDAR 8B drafter	`JetLM/SDAR-8B-Chat-b16`	Cheng et al. (2026a)
SDAR 1.7B drafter	`JetLM/SDAR-1.7B-Chat-b16`	Cheng et al. (2026a)
DFlash drafter	`z-lab/Qwen3-8B-DFlash-b16`	Chen et al. (2026)
DSpark drafter	`deepseek-ai/dspark_qwen3_8b_block7`	Cheng et al. (2026b)

Table 2: Hugging Face checkpoints used in the experiments.

The Dream GSM8K experiment uses the model's chat template without an additional instruction prefix. On GSM8K, Gemma and the Qwen3 comparisons prepend the instruction to put the final answer inside `\boxed{}`. Within each model-pair comparison, the AR baseline uses the corresponding prompt format.

Dream 7B uses entropy-based unmasking with deterministic position selection. Each drafting round denoises the remaining generation canvas.

DiffusionGemma uses a block size of 256. When the DiffusionGemma is run as a standalone model, it uses an entropy bound sampler which terminates the diffusion when a confidence threshold is reached, so the number of diffusion steps is variable, but is bounded by an upper limit. In our experiments, the values of 1, 2, 4, 8, 16, 32, and 48 are used as the upper bound. When DiffusionGemma is run as a drafter, we modify it to run a fixed number of diffusion steps (K) instead of using the original entropy bound sampler, because DiffusionGemma has both a stability threshold and a confidence threshold to determine when to terminate the diffusion, along with the entropy bound sampler threshold, which makes it not straightforward to control the diffusion effort with the original sampling approach.

SDAR commits tokens whose confidence exceeds τ, so its number of denoising passes varies with the draft. We sweep $\tau \in \{0.3, 0.5, 0.7, 0.9\}$ for both model sizes. We also run another configuration where the SDAR drafter runs exactly one diffusion step instead of relying on the confidence threshold for termination. All Qwen3 comparisons use $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ with the same verifier model and sampling settings. Table 4 gives block sizes and completion limits. DSpark retains its native confidence threshold of zero.

Figure 5 uses one-pass Uno drafting. Appendix F.4 reports the additional $K = 2, 4$ experiments that support this choice. For DFlash and DSpark, we vary acceptance leniency while keeping their released drafting procedures.

E.3 Sweep Grids and Measurement Protocol

Table 5 lists the Cactus budgets for the main results, along with the diffusion effort K.

On the other hand, for the linear acceptance modifier, $\lambda \in \{0, 0.1, \ldots, 1\}$ is used, along with $K \in \{1, 2, 3, 4\}$ for the Dream pair on GSM8K, $K \in \{1, 2, 3\}$ for the Dream pair on HumanEval, $K \in \{1, 2, 4, 8\}$ for DiffusionGemma pair on GPQA, and $K \in \{1, 2, 4, 6, 10\}$ for DiffusionGemma pair on HumanEval.

Sampler	Temperature	Top-p	Top-k
Dream draft / standalone	0.2	0.95	–
Qwen2.5-32B verifier	0.7	0.8	20
DiffusionGemma draft / standalone	$0.8 \rightarrow 0.4$	0.95	–
Gemma-4-31B verifier	1.0	0.95	64
SDAR draft	0.6	0.95	20
Uno draft	1.0	0.95	50
DFlash draft	Greedy	–	–
DSpark draft	1.0	–	–
Qwen3-8B verifier / AR	1.0	0.95	50

Table 3: Sampling settings for the reported runs. A dash indicates no additional truncation recorded for that setting. DiffusionGemma anneals temperature during denoising and uses 0.4 for a single denoising step.

Model / pair	Dataset	Diffusion effort	Acceptance	Block	Token limit
Dream 7B + Qwen2.5 32B	GSM8K	$K \in \{1, 2, 3, 4\}$	Cactus	≤ 512	512
	HumanEval	$K \in \{1, 2, 3, 4\}$	Cactus	≤ 512	512
DiffusionGemma 26B-A4B + Gemma-4 31B	GPQA	$K \in \{1, 2, 4, 8\}$	Cactus	256	2048
	GPQA	$K \in \{1, 2, 4, 8\}$	Linear	256	2048
SDAR 8B	GSM8K, HumanEval	$\tau = 0.3, 0.5, 0.7, 0.9$ or fixed $K = 1$	Linear	16	512
SDAR 1.7B	GSM8K, HumanEval	$\tau = 0.3, 0.5, 0.7, 0.9$ or fixed $K = 1$	Linear	16	512
Uno	GSM8K, HumanEval	$K = 1$	Linear	16	512
DFlash	GSM8K, HumanEval	One draft pass	Linear	16	512
DSpark	GSM8K, HumanEval	Fixed sampler	Linear	7	512

Table 4: Main experimental configurations. The final five rows share the Qwen3-8B verifier. Block lengths are the configured values, distinct from full-completion limits. DFlash's 16-position block includes one anchor and 15 draft positions. SDAR's confidence threshold τ controls denoising effort adaptively.

E.4 Simple Throughput Model Calibration

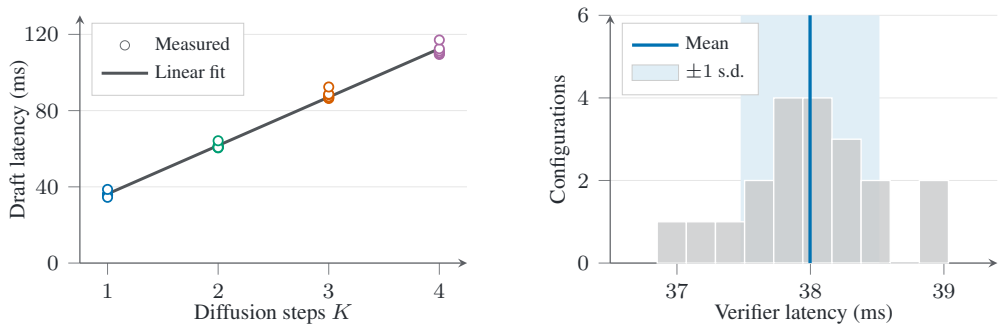
The calibration for Figure 3 uses $K \in \{1, 2, 3, 4\}$ and runs 50 random questions from the GSM8K dataset. For each K, we compute the average TVD between the drafter and verifier distributions. We then substitute them into Equation 6 and derive the theoretical throughput and quality at different leniency values. The block size B is set to 512 in Equation 7.

Based on empirical profiling of the Dream-7B drafter and Qwen2.5-32B verifier, we estimate d_K to be $0.566, 0.517, 0.424, 0.360$ for $K = 1, 2, 3, 4$ respectively. Figure 6 shows the goodness of fit for the linear regression plot and the constant time for calling the verifier.

F Additional Results

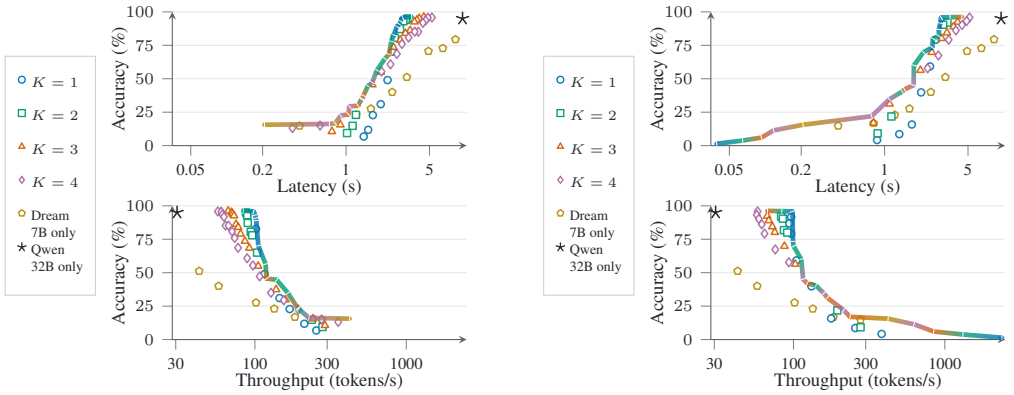
F.1 Linear Acceptance

Figures 7 and 8 show the linear-acceptance counterparts of Figure 4, alongside Cactus for comparison.



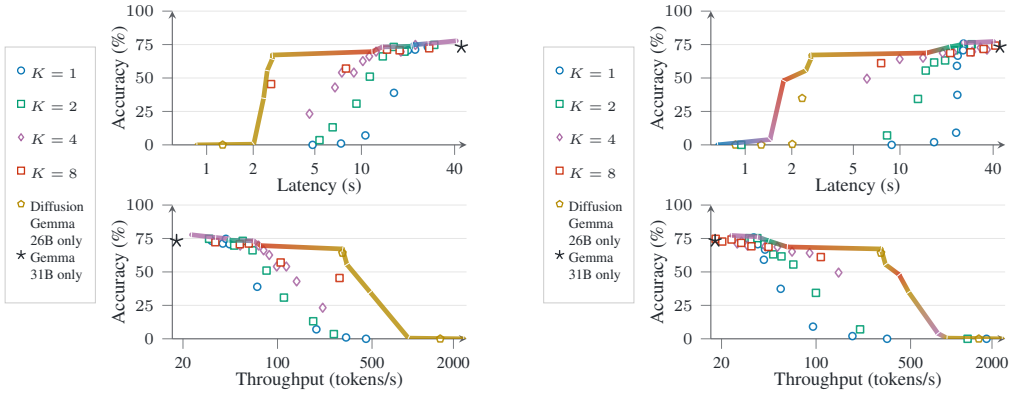
(a) Drafting cost (b) Verification cost

Figure 6: Per-round latency measurements supporting Figure 3. Left: drafting costs and their linear fit in K. Right: the distribution of verifier costs across configurations, with its mean and one-standard-deviation band.



(a) Cactus (b) Linear acceptance

Figure 7: Latency and throughput on GSM8K with the Dream-7B drafter and Qwen2.5-32B-Instruct verifier



(a) Cactus (b) Linear acceptance

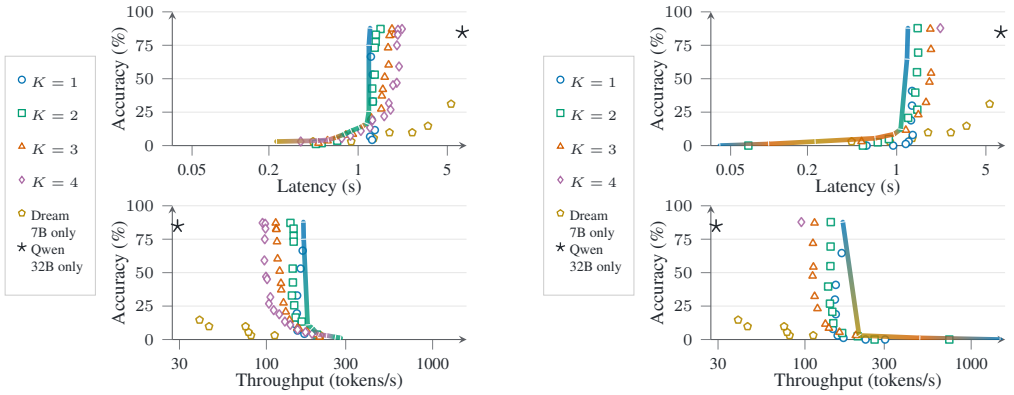
Figure 8: Latency and throughput on GPQA Diamond with the DiffusionGemma 26B drafter and Gemma4 31B verifier.

Model pair / dataset	K	Cactus budgets δ
Dream–Qwen / GSM8K	1	$0, 0.25, 0.5, 1, \ldots, 8$
	2	$0, 0.25, 0.5, 1, \ldots, 11$
	3	$0, 0.25, 0.5, 1, \ldots, 13$
	4	$0, 0.25, 0.5, 1, \ldots, 15$
DiffusionGemma–Gemma / GPQA	1	$0, 0.25, 0.5, 1, \ldots, 5$
	2	$0, 0.25, 0.5, 1, \ldots, 6$
	4	$0, 0.25, 0.5, 1, \ldots, 8, 10$
	8	$1, 3, 5, 7, 10, 15$
Dream–Qwen / HumanEval	1	$0, 0.25, 0.5, 1, \ldots, 6$
	2	$0, 0.25, 0.5, 1, \ldots, 12$
	3	$0, 0.25, 0.5, 1, \ldots, 13$
	4	$0, 0.25, 0.5, 1, \ldots, 15$
DiffusionGemma–Gemma / HumanEval	1	$0, 0.25, 0.5, 1, \ldots, 8$
	2	$0, 0.25, 0.5, 1, \ldots, 12$
	4	$0, 0.25, 0.5, 1, \ldots, 16$
	6	$0, 0.25, 0.5, 1, \ldots, 15$
	10	$0.25, 0.5, 1, \ldots, 15, 20$

Table 5: Evaluated Cactus budgets at each diffusion effort.

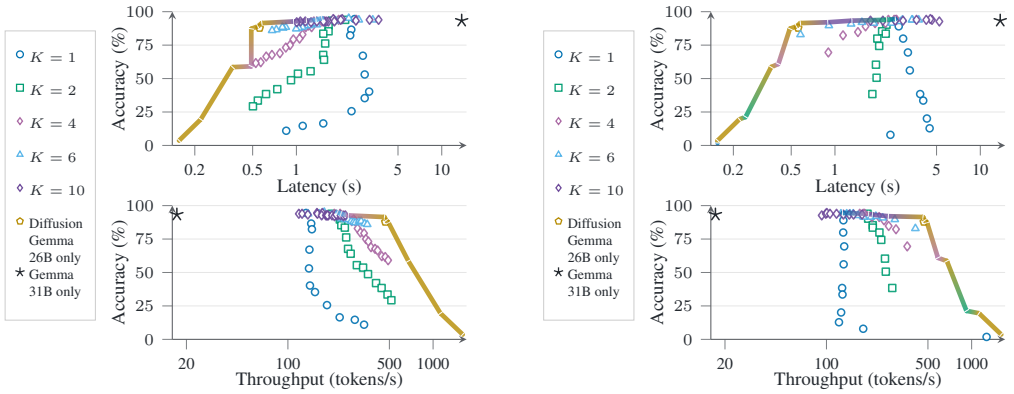
F.2 HumanEval

Figures 9 and 10 extend the Dream–Qwen and DiffusionGemma–Gemma comparisons to HumanEval, using both acceptance rules.



(a) Cactus (b) Linear acceptance

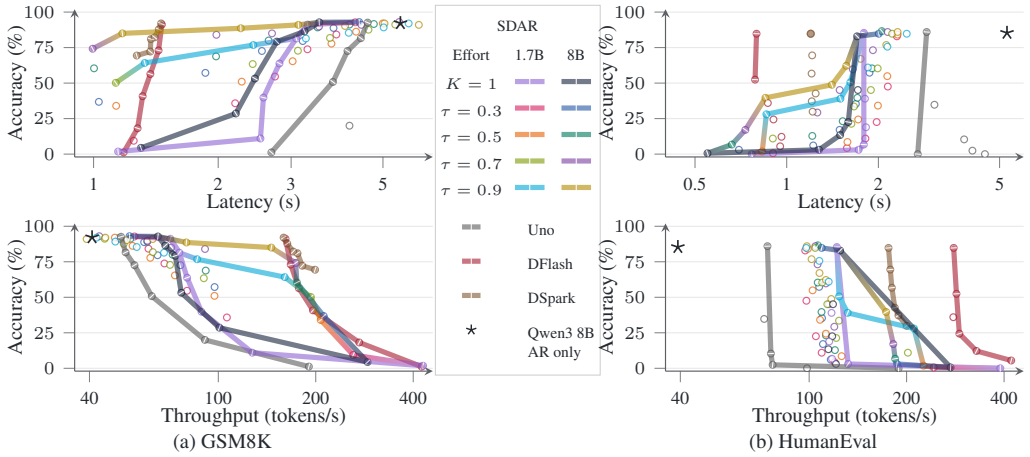
Figure 9: Latency and throughput on HumanEval with the Dream 7B drafter and Qwen2.5 32B verifier.



(a) Cactus (b) Linear acceptance

Figure 10: Latency and throughput on HumanEval with the DiffusionGemma 26B drafter and Gemma4 31B verifier.

F.3 Shared-Verifier Comparison



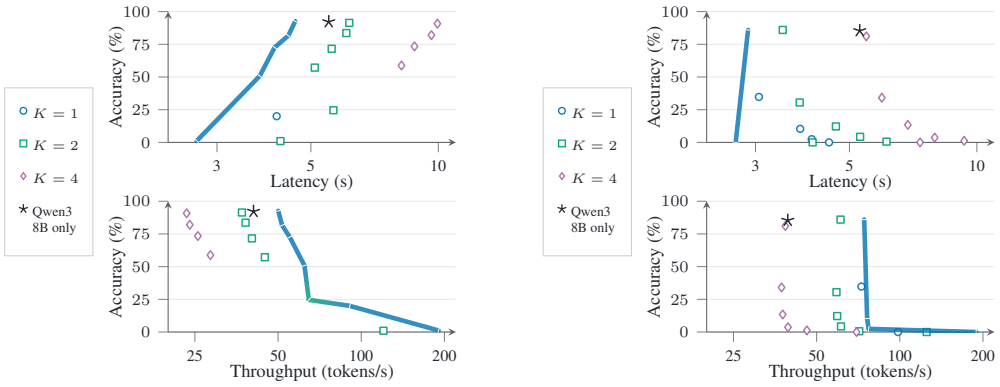
(a) GSM8K (b) HumanEval

Figure 11: Complete sweeps for Figure 5. Colors distinguish SDAR's effort settings within each model size. Hollow markers show measured configurations, including dominated points. Thick curves show the joint and fixed-$K = 1$ frontiers, using the same definitions as the main figure.

F.4 Uno at Multiple Diffusion Steps

We evaluate Uno with $K \in \{1, 2, 4\}$ and linear acceptance at $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. We hold the block size at 16 and use the same problems, sampling settings, and 512-token limit as Figure 5.

In Figure 12, $K = 1$ captures the entire latency frontier on both benchmarks and the entire throughput frontier on HumanEval. On GSM8K, one low-accuracy $K = 2$ configuration also contributes to the measured throughput frontier, while $K = 4$ contributes to neither frontier. Thus, additional steps do not provide the speed-quality frontier improvements seen with diffusion drafters that are trained to perform as standalone models. This pattern is consistent with Uno's one-step distillation objective, which trains it to recover a clean block from fully corrupted inputs (Sahoo et al., 2026) following the base model's distribution.



(a) GSM8K (b) HumanEval

Figure 12: Uno with one, two, and four drafting steps. Colors indicate K, with leniency varied within each color. Thick lines connect nondominated measurements, and hollow markers show the remaining configurations. The GSM8K sweep lacks completed $K = 4$ runs at $\lambda \in \{0.8, 1\}$.