Speculative Diffusion Decoding for Accelerated Language Generation

Jacob K. Christopher
University of Virginia
csk4sr@virginia.edu

Brian R. Bartoldson
Lawrence Livermore National Laboratory
bartoldson1@llnl.gov

Tal Ben-Nun
Lawrence Livermore National Laboratory
talbn@llnl.gov

Michael Cardei
University of Virginia
ntr2rm@virginia.edu

Bhavya Kailkhura
Lawrence Livermore National Laboratory
kailkhura1@llnl.gov

Ferdinando Fioretto
University of Virginia
fioretto@virginia.edu

Abstract

Speculative decoding has emerged as a widely adopted method to accelerate large language model inference without sacrificing the quality of the model outputs. While this technique has facilitated notable speed improvements by enabling parallel sequence verification, its efficiency remains inherently limited by the reliance on incremental token generation in existing draft models. To overcome this limitation, this paper proposes an adaptation of speculative decoding which uses discrete diffusion models to generate draft sequences. This allows parallelization of both the drafting and verification steps, providing significant speedups to the inference process. Our proposed approach, *Speculative Diffusion Decoding (SpecDiff)*, is validated on standard language generation benchmarks and empirically demonstrated to provide **up to 7.2x speedups over standard generation processes and up to 1.75x speedups over existing speculative decoding approaches.**

1 Introduction

As autoregressive language modeling with transformers [Vaswani et al., 2017] is scaled to larger compute levels, performance improves and new capabilities emerge [Kaplan et al., 2020, Brown et al., 2020]. Indeed, scaling has been shown to to improve the performance of large language models (LLMs) for a diverse array of tasks, including code generation, question answering, summarization, and many other use cases [Achiam et al., 2023, Gemini Team, 2023, Llama Team, 2024]. For instance, models such as LLaMA 3.2 90B [Llama Team, 2024], ChatGPT [OpenAI et al., 2024], Cohere 52B [Ruis et al., 2023], Google's Gemini-ULTRA [Team et al., 2024] exemplify the ongoing trend of deploying and releasing increasingly large models, enabling broader access and application across various domains.

However, while these desired capability arise, running LLMs in inference mode for millions of users produces burdensome electricity, time, and monetary demands. Many methods exist to mitigate these costs – including sparsity, quantization, and distillation – but they often introduce new tradeoffs, e.g., their application can degrade the performance of the model [Hong et al., 2024].

Unlike other methods for accelerating LLM inference, *speculative decoding* [Xia et al., 2023, Leviathan et al., 2023] can improve LLM efficiency by 2–3× with *no degradation in the quality of*

4th NeurIPS Efficient Natural Language and Speech Processing Workshop (ENLSP-IV 2024).

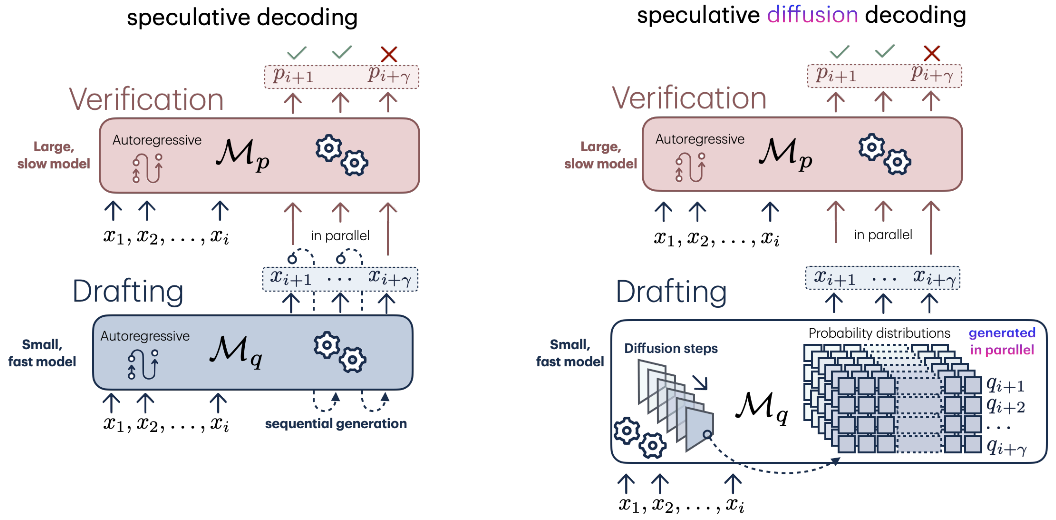

Figure 1: Illustration of classical speculative decoding (left) and speculative diffusion decoding (right).

the model outputs. In Leviathan et al. [2023], speculative decoding achieves this by sequentially generating multiple tokens with a small, efficient draft model, then running the target, large, LLM in parallel on all of the drafted tokens, simultaneously evaluating their consistency with the target LLM's output token probabilities. Provided that the drafting model's tokens are frequently accepted by the target model and that the drafting model operates substantially faster than the target model, speculative decoding can directly match the sampling output from the target model while significantly reducing runtime [Leviathan et al., 2023]. This functioning is shown in Figure 1 (left).

Notably, since both the drafting model's speed and its alignment relative to the target model are critical to the success of speculative decoding, simultaneous improvements in each of these areas are necessary to ensure speculative decoding's relevance to future, more capable target models. For instance, a small GPT-2 [Radford et al., 2019] drafting model could produce drafts that are often rejected by GPT-4 [Achiam et al., 2023], and simply scaling the drafting model to address its weaker generations risks diminishing the speed advantage necessary to speculative decoding's success. To address this challenge, previous efforts have focused on introducing additional parallelization techniques that incorporate prediction trees and branching to refine the drafting process [Fu et al., 2024, Miao et al., 2023a, Svirschevski et al., 2024]. However, the gain in generation efficiency are at the expense of much increased number of operations and/or memory for each generation.

This paper proposes a fundamentally different approach to improve speculative decoding: *It proposes to replace the auto-regressive drafter with recently introduced discrete diffusion models* [Lou et al., 2024, Sahoo et al., 2024]. These models offer several key advantages when used as drafters: Firstly, they provide a smooth trade off between the compute cost of generation and the quality of generation (via the number of reverse diffusion steps). Second, while they have historically struggled relative to traditional language models, recent diffusion models have been shown to require 32× fewer function evaluations than autoregressive models to produce text with comparable perplexity [Lou et al., 2024], with more recent works reporting even further speedups [Sahoo et al., 2024]. This is a trend which is anticipated to continue. Finally, future advances in diffusion model generation quality are highly aligned with their ability to perform strongly as speculative drafters: as drafted tokens are accepted by the target model at a higher rate, a larger number of proposed drafted tokens becomes optimal from an efficiency/speed point of view, and (unlike sequential drafters) diffusion models can easily accommodate generation of many more tokens since they are able to generate entire sequences in a single step.

Contributions. More specifically, this paper makes the following contributions:

1. It introduces a novel integration of generative diffusion language models with speculative decoding, schematically illustrated in Figure 1 (right).

2. It empirically demonstrate the hybrid model's ability to significantly accelerate inference times while maintaining the same high-quality outputs of the original, target large language model.
3. The proposed method ensures that all generations from the diffusion language model, which are empirically shown to produce outputs with significantly higher perplexity than current state-of-the-art autoregressive models [Sahoo et al., 2024, Lou et al., 2024, Austin et al., 2021, Gloeckle et al., 2024], align with the outputs generated by larger, more computationally demanding models.
4. Finally, the paper sets a new benchmark for speed in language completion tasks on the CNN/DM Nallapati et al. [2016]and OpenWebText datasets Gokaslan et al. [2019].

2 Related Work

While autoregressive language models provide state-of-the-art performance on language generation tasks, the incremental decoding used by these architectures results in significant overhead at inference time [Miao et al., 2023b]. This is largely a result of the inability to parallelize the sequential process of generating tokens in the output sequence as each token generation is dependent upon the preceding tokens in the sequence; consequentially, scaling the compute associated with the inference cannot directly reduce this overhead when using standard decoding schemes. In recent literature studying how to accelerate large language model generation, two primary approaches have been explored: (1) advanced decoding implementations that better parallelize token generation and (2) non-autoregressive language models allowing full sequences to be generated simultaneously.

Speculative decoding. Speculative decoding accelerates autoregressive generation by leveraging a smaller autoregressive models of the same architecture (the drafter model) to predict candidate sequences, which the original model (the target model) then verifies [Leviathan et al., 2023, Chen et al., 2023]. Notably, the earliest literature on speculative diffusion adapted a non-autoregressive model to act as the drafter model [Xia et al., 2023], using a masked language model with a bidirectional decoder [Ghazvininejad et al., 2019]. However, the integration of non-autoregressive draft models has not received much attention due to the difficulty introduced by the necessary additional training in existing approaches and the modest speedups that were previously reported using these methods (less than 2x speedup over vanilla decoding schemes).

Thus, recent advancements in speculative decoding have focused on overcoming memory-related constraints, with improvements achieved through various approaches: drafting directly with the target model [Cai et al., 2024, Zhang et al., 2024], enhancing draft algorithms [Sun et al., 2024], and introducing additional parallelization techniques that incorporate branching to refine the drafting process [Fu et al., 2024, Miao et al., 2023a, Svirschevski et al., 2024].

Non-autoregressive language models. Models which stray from the autoregressive paradigm have been shown to speedup generation by generating blocks or even entire sequences simultaneously. Gloeckle et al. [2024] propose a method of adapting traditional autoregressive models to sample blocks of tokens, improving inference time over similarly scaled models. In a similar vein, diffusion language models have been recognized for their efficiency in generating extended token sequences concurrently, offering even greater speed enhancements. These models recast language generation as a diffusion process either across the embedding space [Austin et al., 2021] or, more recently, through the probability distributions of generated tokens [Sahoo et al., 2024, Lou et al., 2024]. Recent models report up to a 32x speedup over similarly sized GPT-2 models Lou et al. [2024], and current state-of-the-art further improves runtime speed Sahoo et al. [2024]. However, despite they have been shown to dramatically accelerate the inference time for language generation, diffusion models typically perform less effectively than state-of-the-art autoregressive models in terms of standard language metrics, often exhibiting significantly higher perplexity scores Zheng et al. [2024]. In the following section, *we will demonstrate, for the first time, how the speed of these models can be leveraged without being subject to this critical limitation.*

3 Preliminaries and Settings

For open-ended language generation, we focus on the task of token generation, where given a sequence of tokens $x_1, x_2, \ldots, x_i$, denoted here with shorthand notation $x_{1:i}$, the goal is to generate the next

n tokens $x_{i+1}, \ldots, x_{i+n}$ from the conditional distributions $p(x_{i+1}|x_{1:i}), \ldots, p(x_{i+n}|x_{1:i+n-1})$ or more succinctly $p_{i+1}, \ldots, p_{i+n}$.

Speculative decoding. Speculative decoding leverages two LLMs, M_p and M_q, to parallelize token generation:

- M_p is the original, *target*, model whose output probability distributions for the tokens are $p_{i+1}, \ldots, p_{i+n}$.
- M_q is a smaller and more efficient *drafter* model, used to generate approximations of the distribution of M_p as $q_{i+1}, \ldots, q_{i+n}$.

This process follows a *draft-then-verify* approach [Stern et al., 2018], where M_q efficiently computes a candidate sequence of tokens, which M_p then verifies in parallel.

During each speculative decoding iteration, M_q generates a subset of the total n tokens that are required for the generation task. The size of this subset is denoted as γ. As shown in Figure 1 (left), the tokens $x_{i+1:i+\gamma}$ sampled from M_q are then used by M_p to generate the corresponding probability distributions $p_{i+1}, \ldots, p_{i+\gamma}$. The distributions $q_{i+1}, \ldots, q_{i+\gamma}$ from M_q are stored for evaluating acceptance in subsequent steps. Critically, the target model's inference over $p_{i+1}, \ldots, p_{i+\gamma}$ can now be run in parallel as the model has access to tokens $x_{i+1:i+\gamma}$, alleviating the sequential dependency for generation with M_p.

To ensure high-quality outputs despite potential discrepancies between M_p and M_q, tokens are subjected to an acceptance criterion. For each token x_j with $j \in [i+1, i+\gamma]$, if $q(x_j) \leq p(x_j)$, the token is accepted. If $q(x_j) > p(x_k)$, the token is rejected with a probability of $1 - \frac{p(x_j)}{q(x_j)}$. This criterion is applied sequentially from left to right; rejection of any token results in the discard of all subsequent tokens. Hence, the token acceptance is maximized when the output distributions of M_q and M_p are closely aligned.

Previous literature quantifies the likelihood of token acceptance, denoted α, and theoretically demonstrate that $\alpha = 1 - \mathbb{E}(D_{LK}(p, q))$ where D_{LK} represents a divergence measure between distributions Leviathan et al. [2023]. This has led to the prevalent use of drafters taken from the same series as the target models, *a paradigm that we challenge in this paper*.

4 Speculative Diffusion Models

Speculative decoding has provided state-of-the-art results for improving language generation inference time but requires meticulous tuning of the associated hyperparameters to achieve optimal results. Particularly γ, the sequence length generated by the drafter model, needs to be appropriately calibrated not only to maximize potential speedup but to even outperform standard autoregressive decoding. This is an important consideration when using current autoregressive draft models, provided that the inference time to generate $M_q(x)$, the draft logits, is directly scaled by the size of γ. Increasing this value too high reduces the number of operations that are conducted in parallel, *potentially leading to speculative decoding increasing inference time,* while reducing this value too low results in speculative decoding "missing out" on token generations that could have been handled by the draft model.

Leviathan et al. [2023] has conducted theoretical analysis on how to best optimize the value of γ, however, it has been contingent upon accurately estimating the percentage of tokens in a the sequence that will be accepted by the target model. By their own acknowledgment, it would be necessary to predict this value*for each draft* and numerically solve for the optimal value of γ to fully realize the potential speedup of speculative decoding. Thus, a significant portion of the residual suboptimality in current implementations can be attributed directly to the sensitivity of this hyperparameter.

Diffusion language models are juxtaposed to conventional language models in that they do not sample token sequences in a sequential manner, rather generating entire sequences in parallel. This has resulted in significant speedup over similarly sized autoregressive models when generating extended sequences [Lou et al., 2024]. This can particularly be observed in longer sequence generations as scaling the draft length γ results in minimal overhead due to the ability to directly parallelize token generation.

4.1 SpecDiff: Formulation

Diffusion models generally operate on continuous data spaces by progressively adding Gaussian noise to the data in a forward diffusion process and then learning to reverse this process to generate new samples Sohl-Dickstein et al. [2015a], Ho et al. [2020], Song and Ermon [2019]. This framework is well-suited for data types such as images, where pixel values can be treated as continuous and thus can naturally accommodate additive Gaussian noise. However, when dealing with discrete data like natural language (tokens), the assumption of continuous noise addition does not hold, as it would result in non-integer values that do not correspond to valid tokens.

Discrete diffusion models address this limitation by redefining the forward processes to transition between discrete token states, such as replacing tokens according to a transition matrix or introducing randomness through categorical distributions Austin et al. [2021], Hoogeboom et al. [2021], Sohl-Dickstein et al. [2015b]. This process enables the generation of coherent and meaningful sequences in natural language processing tasks. However, unlike continuous diffusion models, traditional score-matching techniques cannot be directly applied to learn discrete diffusion models. Instead, various surrogate losses have been proposed for training.

In particular, Sahoo et al. introduce the Masked Diffusion Language Model (MDLM), which gradually masks and then reconstructs tokens within a text sequence, enabling efficient text generation. MDLM optimizes a continuous-time Negative ELBO (NELBO) objective to minimize the negative log-likelihood over a continuous-time diffusion process, which can be formulated as follows:

$$L_{\infty}^{\text{NELBO}} = \mathbb{E}_{\tilde{q}} \left[\int_{t=0}^{t=1} \frac{(1-\beta_t)'}{\beta_t} \sum_i \log \langle x_\theta(z_i^t), x_i \rangle \, dt \right]$$

where $x_\theta(z_i^t)$ represents the model's estimate of the original token x_i at time t given the current noisy state z_i^t, and β_t denotes the noise schedule controlling the diffusion process. Here, $\tilde{q}$ is the forward noising process of in the masked diffusion, defining the distribution over the noisy latent variable z_i^t, and can be related to the NELBO as described in Equation (8) of Sahoo et al., 2024. The expectation $\mathbb{E}_{\tilde{q}}$ signifies averaging over the possible outcomes of $\tilde{q}$ allowing the model to accound for all possible variations of z_t. The Noise Schedule Derivative term, $\frac{(1-\beta_t)'}{\beta_t}$, represents the rate of change in the noise schedule β_t over time, and t is a continuous timestep between 0 and 1.

This loss is directly used for pretraining our draft model. As this process learns to denoise over the probability mass vectors, the output of the draft model is a matrix of $\mathbb{R}^{n \times m}$ where $n = \gamma$ and m is the size of the vocabulary. The candidate sequence can then be generated using standard decoding methods over these probability mass vectors, the logits of which are stored to determine whether to accept each draft token.

Now, the draft logits produced by the output matrix of the discrete diffusion drafter directly substitute the autoregressive drafter used to generate $M_q([x_{0:i}] \circ [x_{i+1}, \ldots, x_{i+\gamma}])$, where $\circ$ is a list concatenation operator. This substitutes the draft step taken by Leviathan et al. and Chen et al. and is the primary difference between SpecDiff and standard speculative decoding approaches. The subsequent steps of verifying this draft with the target model follow the previously proposed decoding algorithm, thus our proposal requires minimal modifications to existing speculative decoding code bases.

Algorithm 1: SpecDiff Decoding

```
▷ Take $T$ diffusion steps to generate the draft.
$q^T_{i+1,\ldots,i+\gamma} \sim \mathcal{N}(\mathbf{0}, \sigma_T \mathbf{I})$
for $t = T$ to 1 do
    $q^{t-1}_{i+1,\ldots,i+\gamma}(x) \leftarrow$
        $M_q([x_0, \ldots, x_i] + [q^t_{i+1,\ldots,i+\gamma}(x)], t)$
$x_{i+1,\ldots,i+\gamma} \sim q^0$
▷ Run $M_p$ in parallel.
$p_i(x), \ldots, p_{i+\gamma+1}(x) \leftarrow$
    $M_p(x_0, \ldots, x_i), \ldots, M_p(x_0, \ldots, x_{i+\gamma})$
▷ Determine the number of accepted guesses $n$.
$r_i \sim U(0,1), \ldots, r_{i+\gamma} \sim U(0,1)$
$n \leftarrow \min(\{j - 1 \mid i \leq j \leq i + \gamma, r_j > \frac{p_j(x)}{q_j(x)}\} \cup \{\gamma\})$
▷ Adjust the distribution from $M_p$ if needed.
$p'(x) \leftarrow p_{n+1}(x)$
if $n < i + \gamma$ then
    $p'(x) \leftarrow$
        $norm(max(0, p_{n+1}(x) - q_{n+1}(x)))$
▷ Return one token from $M_p$, $n$ tokens from $M_q$.
$t \sim p'(x)$
return $x_1, \ldots, x_n, t$
```

A complete overview of the SpecDiff decoding is provided in Algorithm 1 (adapted from Leviathan et al.).

Drafter's evaluations. Next, we highlight an important difference between standard speculative decoding and our approach. While in standard speculative decoding the number of evaluations by the drafter model is dictated by the value of γ (used in the first loop for Algorithm 1), in speculative diffusion it is dictated by the number of diffusion steps, T. This allows SpecDiff to scale γ to higher values, as discussed further in Section 6.

Instead, the value of T is selected to optimize the trade-off between draft quality and computational overhead. While analysis by Lou et al. shows that lower values of T lead to higher perplexity in the generated sequence, this only impacts SpecDiff with respect to its effect on the percentage of tokens from the draft which are accepted (as can be noted in the analysis reported in Figure 2 and discussed in details Section 6).

Sequence initialization. An important consideration in implementing SpecDiff is the initial alignment between the diffusion draft model and the target model's data distribution since these models architectures are fundamentally different one another. A key strength of SpecDiff lies in its ability to leverage the alignment between the prefixes used in discrete diffusion and the target model's data distribution. Specifically, the better the prefixes align with the target distribution, the more effectively the diffusion drafter can generate longer, coherent sequences matching the target distribution, resulting in progressively higher acceptance rates.

This however, also means that when the diffusion draft model has not been finetuned, its output distribution may initially differ from that of the target model, potentially leading to lower acceptance rates at the beginning of the generation process. To address this, we employ standard speculative decoding for the initial few tokens, thereby optimizing SpecDiff's performance from the outset. This strategy ensures that SpecDiff achieves speedups that are empirically at least as significant as those observed with standard speculative decoding and can realize substantial improvements, as we show in Tables 1 and 6. Moreover, once the diffusion draft model is finetuned to better match the target distribution, this initial speculative decoding becomes unnecessary, as the drafter effectively aligns with the target model from the beginning.

5 Evaluation

To empirically evaluate the improvements provided by using SpecDiff, the paper provides an empirical analysis on text summarization and more general text generation, leveraging benchmarks that are common to existing literature. Additionally, we assess the performance of our method against the current state-of-the-art using *SpecBench*, a unified evaluation platform for speculative decoding techniques Xia et al. [2024]. All evaluation is conducted on two NVIDIA A100 series GPUs (80GB) using CUDA 12.2. Additionally, FlashAttention Dao et al. [2022] is used to optimize the performance in all experiments.

5.1 Experimental Setup

Settings. Evaluation is conducted on three standard natural language processing tasks: (1) text summarization using the CNN/DM dataset Nallapati et al. [2016], (2) text generation on the OpenWebText (OWT) dataset Gokaslan et al. [2019], and (3) text generation using MT Bench Zheng et al. [2023]. In each setting the model is queried for 1024 tokens using a greedy decoding scheme (temperature = 0). For the experiments, we evaluate the pretrained target models M_p GPT-2 XL (1.5B), GPT-NEO (2.7B), and Vicuna (33B). We use Masked Diffusion Language Model (110M) as our drafter model M_q, which is a comparable size to the baseline drafter GPT-2 (86M) employed for our standard speculative decoding baseline Sahoo et al. [2024].

Evaluation metrics. Our method is assessed empirically by walltime speedup and acceptance rate α. The reported results are compared to recognized baselines of vanilla autoregressive decoding, standard speculative decoding (SpS) implementations as proposed by Leviathan et al., Chen et al., which our method most closely resembles. Additionally, we provide comprehensive comparison to these method, as well as the current fastest-to-date speculative decoding approaches, detailing the improvements SpecDiff provides in reduction of FLOPs and memory footprint in Table 3.

6 Results and Discussion

Empirically we highlight the comparison between our approach, other speculative decoding methods, and vanilla autoregressive generation. Across the tested settings and target model architectures, SpecDiff significantly outperforms standard speculative decoding, achieving speedups of up to 7.2x compared to the target models and **increasing the efficiency of standard speculative decoding by more than 1.75x**. Furthermore, SpecDiff matches the current state-of-the-art methods while **reducing the FLOPs/draft by over 33$\%$ and reducing memory consumption (shown in Table 3).**

6.1 Comparison with Standard Speculative Decoding

First, we compare SpecDiff to the performance of standard speculative decoding in Table 1. Despite SpS generally reporting slightly higher acceptance rates, the **improved parallelization provided by SpecDiff results in greater speedups ranging from 1.45–$1.75\times$ improvement over the baseline.** As we have not finetuned the target model or draft model, we utilize sequence initialization with SpS to enhance generation speed by sampling the first 100 tokens of the sequence. We ablate the performance of SpecDiff without this enhancement in the Table 6.

		M_p	M_q	γ	α	Speedup
CNN/DM	**SpS**	GPT-2 XL	GPT-2	8	0.92	3.58x
		GPT-NEO	GPT-2	9	0.95	5.45x
	Ours	GPT-2 XL	MDLM	15	0.87	**4.80x**
		GPT NEO	MDLM	15	0.88	**6.63x**
OpenWebText	**SpS**	GPT-2 XL	GPT-2	8	0.93	3.66x
		GPT-NEO	GPT-2	7	0.85	4.12x
	Ours	GPT-2 XL	MDLM	15	0.89	**5.38x**
		GPT NEO	MDLM	20	0.88	**7.23x**

Table 1: Evaluation of walltime speedup over autoregressive decoding using SpecDiff (Ours) compared to standard speculative decoding (SpS). The best result for each setting and target model is displayed in **bold**.

6.2 Evaluation Against State-of-the-Art Speculative Decoding Methods

To provide comparison to other state-of-the-art speculative decoding methods, evaluation is conducted on generation prompts from MT Bench Zheng et al. [2023] using Vicuna 33B as the target model. Generation lengths are reduced to 512 tokens to better align with the task. For this setting, MDLM (141M) is pretrained with the Llama 1 tokenizer for compatibility with the target model used. As Vicuna is instruction tuned, we utilize a teacher-student instruction tuning approach resembling the knowledge distillation approach proposed by Zhou et al. to align the output distribution of MDLM with the target model. This approach is similar to the additional instruction tuning conducted to align Eagle and Eagle-2 on ShareGPT instruction tuning dataset Li et al. [2024a,b]. Because of this finetuning step, it is not necessary to use sequence initialization as with pretrained models. **Notably, SpecDiff provides a speedup identical to the state-of-the-art method Eagle-2, while requiring 33$\%$ fewer floating point operations per draft generation.** Thorough comparison to Eagle-2 and other speculative decoding strategies is highlighted in Table 3.

		M_p	M_q	γ	α	FLOPs/draft	Speedup
MT Bench	**EAGLE**†	Vicuna 33B	AR Head (990M)	$\approx$5	0.80	N/A	2.41x
	EAGLE-2†	Vicuna 33B	AR Head (990M)	$\approx$6	0.84	8.35×10^{10}	**2.60x**
	Ours	Vicuna 33B	MDLM (141M)	15	0.76	5.53×10^{10}	**2.61x**

Table 2: Evaluation on MT Bench using SpecDiff (Ours), EAGLE, EAGLE-2, and SpS with Vicuna target models. FLOPs/step computes that floating point operations for the drafter to generate each sequence. † denotes results collected using *SpecBench* Xia et al. [2024].

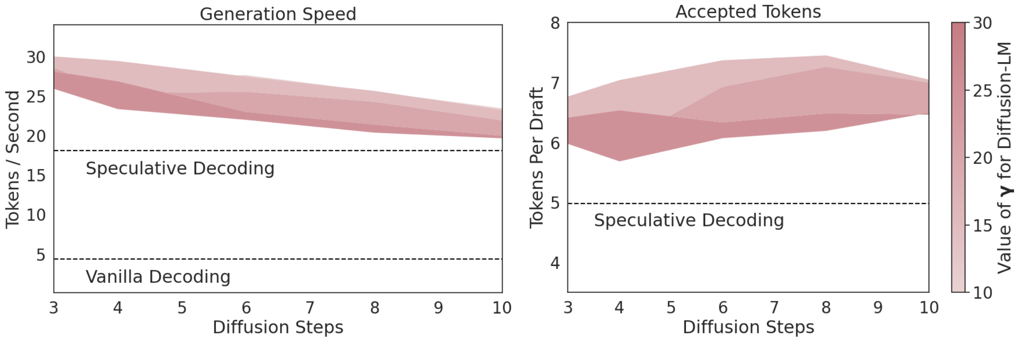

Figure 2: Evaluation of SpecDiff's sensitivity to γ and number of diffusion steps when optimizing speed (left) and accepted tokens per draft (right) as reported on the OpenWebText task using GPT-2 NEO as the target model. Average token acceptance increases with T (x-axis), as examined theoretically in Appendix B. Despite this, the additional compute required as T is scaled results in reduced speedup.

6.3 Robustness to Diverse Drafter Architectures

We finally make an important remark: while previous implementations of speculative decoding rely on a common architecture between the drafter and target models Leviathan et al. [2023], Chen et al. [2023], using smaller versions of the same architecture to generate draft sequences, these experiments demonstrate a robustness to using a completely different architecture for sequence drafting. *This is particularly significant given the absence of finetuning in the reported results (Tables 1 and 6).* Pretrained diffusion models can be directly purposed as draft models requiring no additional training.

The much larger values of γ used for SpecDiff should particularly be highlighted. This is a key discrepancy between diffusion language models, which generate entire sequences in parallel, and autoregressive models. Hence, there is minimal overhead to increasing the sequence length generated by the diffusion-based drafter, and γ can be significantly increased without incurring significant cost.

The hyperparameters used in the reported results have been optimized empirically. We highlight that while in standard speculative diffusion the performance is highly sensitive to γ, SpecDiff is robust to a range of values for γ making it unnecessary to precisely tune this hyperparameter (in our experiments we found between 10 and 20 worked well). Rather, SpecDiff's performance is much more sensitive to the number of diffusion steps selected. Similar to the role of γ in an autoregressive model, the number of diffusion steps T dictates the number of network evaluations during a single drafting step. As reported in the Figure 2, while increasing this hyperparameter arbitrarily results in higher values of α, SpecDiff performs best when this is optimized to balance the objectives of maximizing the number of accepted tokens and minimizing the drafter's overhead.

7 Limitations and Future Work

While our proposed Speculative Diffusion Decoding method represents a significant step forward in accelerating large language model inference, several limitations warrant discussion. Addressing these limitations highlights promising avenues for future research that could further enhance the efficiency and applicability of SpecDiff.

Calibration of discrete diffusion models. A primary limitation of our approach lies in the challenge of using different architectures for the drafter and verifier models. Specifically, the adopted discrete diffusion models do not output well-calibrated probability distributions that align with those of the target autoregressive models, particularly when the sampling temperature (T) is greater than zero. The diffusion models tend to produce over-confident predictions, often assigning near-certain probability to the top-1 token while assigning negligible probabilities to all other tokens. This results in deterministic sampling regardless of the temperature setting, challenging the applicability of SpecDiff in scenarios where diversity and stochasticity in generation are desired.

Thus, the development of techniques to better align the output probabilities of diffusion drafters with the target models is an key area of future work. Achieving proper calibration would enable effective use of SpecDiff at higher temperatures, as well as unlocking massive further speedups beyond what we have reported, as the acceptance rates are predicted to increase dramatically.

Limited tokenization availability. We note that available discrete diffusion models are based on the GPT-2 tokenizer, which would restrict one immediate compatibility with target models using different tokenization schemes. For our experiments, we indeed trained a new discrete diffusion model from scratch with a different tokenizer, a process that demanded substantial computational resources and time. All our models will be released on HuggingFace and thus be directly used by the community.

Performance on shorter generation tasks. We observe that SpecDiff exhibits optimal performance on longer sequence generations. In shorter generation tasks, the benefits of parallelization are less pronounced, and finetuning the diffusion drafter may be necessary to achieve comparable efficiency gains. Without finetuning, the drafter may not effectively capture the target model's token distributions for shorter sequences.

An interesting outcome of these observations is that tailoring the drafter to better model shorter sequences, could improve its alignment with the target model, thereby maintaining speedups even in less extensive generation tasks. We believe that this adjustment may broaden SpecDiff's applicability.

Stochastic sampling. Note that our experiments are conducted with the sampling temperature set to zero, resulting in deterministic token generation. While this setting simplifies the verification process it limits the exploration of the model's capabilities in generating diverse outputs. In the future, we plan to exploring SpecDiff's performance at non-zero temperatures. As highlighted earlier, addressing the calibration issue of the diffusion drafter would enable effective stochastic sampling.

Despite these challenges, our work lays the foundational framework for integrating discrete diffusion models with autoregressive models in speculative decoding. Each limitation discussed highlights a specific area where further research could yield significant benefits for the performance of the proposed SpecDiff. We are optimistic that overcoming these challenges will not only reinforce the strengths of SpecDiff but also unlock new possibilities for accelerating language model inference.

8 Conclusion

Motivated by the costly inference time of current large language models, this paper has proposed the novel integration of discrete diffusion models with autoregressive language models. The proposed method, Speculative Diffusion Decoding, alters existing speculative decoding schemes to integrate a non-autoregressive diffusion model as the draft model. As shown by the empirical evaluation on standard language generation benchmarks, the proposed method leverages the dramatic runtime advantages of discrete diffusion models while also maintaining the dramatically higher generation quality of autoregressive target models. The reported results demonstrate the utility of this approach in effectively accelerating runtime, outperforming vanilla decoding by over 7x and speculative decoding methods by over 1.75x.

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Gemini Team. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Llama Team. The llama 3 herd of models. 2024.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie

Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL `https://arxiv.org/abs/2303.08774`.

Laura Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. The goldilocks of pragmatic understanding: Fine-tuning strategy matters for implicature resolution by llms, 2023. URL `https://arxiv.org/abs/2210.14986`.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur,

Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy,

Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2024. URL `https://arxiv.org/abs/2312.11805`.

Junyuan Hong, Jinhao Duan, Chenhui Zhang, LI Zhangheng, Chulin Xie, Kelsey Lieberman, James Diffenderfer, Brian R Bartoldson, AJAY KUMAR JAISWAL, Kaidi Xu, et al. Decoding compressed trust: Scrutinizing the trustworthiness of efficient llms under compression. In *Forty-first International Conference on Machine Learning*, 2024.

Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3909–3925, 2023.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. Break the sequential dependency of llm inference using lookahead decoding. *arXiv preprint arXiv:2402.02057*, 2024.

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 generative large language model serving with tree-based speculative inference and verification. *arXiv preprint arXiv:2305.09781*, 2023a.

Ruslan Svirschevski, Avner May, Zhuoming Chen, Beidi Chen, Zhihao Jia, and Max Ryabinin. Specexec: Massively parallel speculative decoding for interactive llm inference on consumer devices. *arXiv preprint arXiv:2406.02532*, 2024.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*, 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. *arXiv preprint arXiv:2406.07524*, 2024.

John Austin, David D. Johnson, Jonathan Ho, David Tarlow, and Remco van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/XXXX/XXXX`.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction, 2024. URL `https://arxiv.org/abs/2404.19737`.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus, 2019.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Hongyi Jin, Tianqi Chen, and Zhihao Jia. Towards efficient generative large language model serving: A survey from algorithms to systems. *arXiv preprint arXiv:2312.15234*, 2023b.

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.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*, 2019.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. Medusa: Simple llm inference acceleration framework with multiple decoding heads, 2024. URL `https://arxiv.org/abs/2401.10774`.

Aonan Zhang, Chong Wang, Yi Wang, Xuanyu Zhang, and Yunfei Cheng. Recurrent drafter for fast speculative decoding in large language models, 2024. URL https://arxiv.org/abs/2403.09919.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. Spectr: Fast speculative decoding via optimal transport, 2024. URL https://arxiv.org/abs/2310.15141.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.

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

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015a.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Erik Hoogeboom, Niels Bonnier, Tal Schuster Cohen, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, volume 34, pages 12454–12465, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/XXXX/XXXX.

Jascha Sohl-Dickstein, Eric Weiss, Nikhil Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265, 2015b. URL https://proceedings.mlr.press/v37/sohl15.html.

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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics ACL 2024*, pages 7655–7671, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.findings-acl.456. URL https://aclanthology.org/2024.findings-acl.456.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. Distillspec: Improving speculative decoding via knowledge distillation, 2024. URL https://arxiv.org/abs/2310.08461.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *International Conference on Machine Learning*, 07 2024a.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-2: Faster inference of language models with dynamic draft trees. In *Empirical Methods in Natural Language Processing*, 11 2024b.

Robert D. Blumofe and Charles E. Leiserson. Scheduling multithreaded computations by work stealing. *J. ACM*, 46(5):720–748, September 1999. ISSN 0004-5411. doi:10.1145/324133.324234. URL https://doi.org/10.1145/324133.324234.

Apoorv Saxena. Prompt lookup decoding, November 2023. URL https://github.com/apoorvumang/prompt-lookup-decoding/.

Zachary Ankner, Rishab Parthasarathy, Aniruddha Nrusimha, Christopher Rinard, Jonathan Ragan-Kelley, and William Brandon. Hydra: Sequentially-dependent draft heads for medusa decoding, 2024. URL https://arxiv.org/abs/2402.05109.

Gen Li, Yuting Wei, Yuxin Chen, and Yuejie Chi. Towards faster non-asymptotic convergence for diffusion-based generative models. *arXiv preprint arXiv:2306.09251*, 2023.

Method	**Total Memory Consumption**	**FLOPs**	**Depth**	**Max Tokens**	**Extra Training Requirements**
Autoregressive	$C_p(1)$	$F_p(1)$	D_p	1	N/A
SpS Leviathan et al. [2023][1]	$C_p(\gamma) + C_q(\gamma)$	$F_p(\gamma) + F_q(\gamma)$	$D_p + \gamma D_q$	$1 + \gamma$	M_q
PLD Saxena [2023]	$C_p(\gamma)$	$F_p(\gamma) + \mathcal{O}(Pn)$	$D_p + \mathcal{O}(n)$	$1 + \gamma$	None
Lookahead Fu et al. [2024][1]	$C_p\left(2\gamma(n-1)\right)$	$F_p\left(2\gamma(n-1)\right) + F_p(1)$	$D_p + \mathcal{O}(1)$	$n - 1$	None
Medusa-1 Cai et al. [2024]	$C_p(\mathcal{T}_n) + KC_q(\gamma)$	$F_p(\mathcal{T}_n) + KF_q(\gamma)$	$D_p + D_q$	$1 + \mathcal{T}_d$	M_q
Medusa-2 Cai et al. [2024][2]	—	—	—	—	M_p
Hydra Ankner et al. [2024][3]	—	—	$D_p + KD_q$	—	M_q
EAGLE Li et al. [2024a]	$C_p(25) + C_q(25)$	$F_p(25) + 5F_q(5)$	$D_p + 5D_q$	6	M_q
EAGLE-2 Li et al. [2024b][4]	$C_p(\mathcal{T}_n) + C_q(\mathcal{T}_n)$	$F_p(\mathcal{T}_n) + \mathcal{T}_d F_q(\frac{\mathcal{T}_n}{\mathcal{T}_d})$	$D_p + \mathcal{T}_d D_q$	$1 + \mathcal{T}_d$	M_q
SpecDiff (Ours)	$C_p(\gamma) + C_q(\gamma)$	$F_p(\gamma) + TF_q(\gamma)$	$D_p + TD_q$	$1 + \gamma$	M_q

Table 3: Comparison of different speculative decoding strategies and their parallel properties. The quantities represent the cost of one additional decoding step (i.e., following the initial prompt). Note that M_q (and related quantities such as F_q) are not constant across methods; e.g., EAGLE uses a different draft model architecture than SpecDiff. T is the number of diffusion steps, $\mathcal{T}$ is a sparse draft tree with $\mathcal{T}_n$ nodes and depth $\mathcal{T}_d$, K is the number of heads in a multi-head speculative decoder, P is the number of prompt tokens, n is an n-gram length, and γ is the number of proposed tokens. [1] $\mathcal{O}(1)$ represents a database lookup. [2] Medusa-1 with target fine-tuning for Medusa-2. [3] Medusa with sequential draft heads for Hydra. [4] EAGLE with dynamic draft trees for EAGLE-2.

A Comparative Analysis of Speculative Decoders

Table 3 compares the computational aspects of SpecDiff with state-of-the-art baselines in speculative decoding.

According to the Work-Depth parallel computation model Blumofe and Leiserson [1999], an algorithm can be represented by a DAG, in which each node is an operation and each edge is a dependency. The longest shortest path in this computational DAG (i.e., the longest dependency chain) is called the *depth*. It follows that the average parallelism of a computation is the total number of nodes divided by the depth, and that higher depth means fewer opportunities for parallelism. We represent the operations by the FLOPs of the model.

The table reasons about the approaches through three parameters for M_p and M_q: C (memory consumption), F (arithmetic/floating-point operations, or FLOPs), and D (depth). Due to the quadratic memory and computation requirements of transformers, and for simplicity, $C_{\{p,q\}}$ and $F_{\{p,q\}}$ are represented as functions of the number of tokens after the initial prompt, i.e., $C_p(k)$ is the memory cost for generating $P + k$ tokens, where P is the number of prompt tokens. The depth $D_{\{p,q\}}$ is generally independent of the number of tokens computed and is thus represented by a scalar.

In Medusa, the prediction tree is set by a number of fixed hyperparameters Cai et al. [2024], which we aggregate by using the number of nodes and the depth of the tree $\mathcal{T}_n$ and $\mathcal{T}_d$. Medusa-2 exhibits the same parallel properties as Medusa-1, but requires a joint fine-tuning of M_p along with M_q. Hydra is an adaptation of Medusa, in which the heads are applied in sequence. It thus only affects the depth of the computations.

EAGLE uses a drafter model that contains the target embedding layer, one autoregressive layer, and the target LM head. The fixed tree size is 5 levels deep and contains 25 nodes Li et al. [2024a]. EAGLE-2 Li et al. [2024b] dispenses with the fixed tree defined in EAGLE, and the induced dynamic tree used in generation is represented with $\mathcal{T}$.

Notice that in our method, the depth of the algorithm is dependent on T, the number of diffusion steps, rather than γ. Empirically, T is smaller than γ, which enables more parallelism in the computation w.r.t. the number of generated tokens. Combined with diffusion models enabling longer prediction horizons, this allows SpecDiff to produce more speculative predictions faster.

Practically, we observe that this enables SpecDiff to produce near state-of-the-art inference acceleration while requiring significantly less computational overhead and reduced memory requirements.

B Drafter Convergence Analysis

In previous speculative decoding approaches the computational overhead of the drafting stage has been proportional to γ, as this parameter dictates the number of network evaluations during the draft phase. When using a discrete diffusion draft model, the number of network evaluations is dictated by the number of diffusion steps T. The following reports an analysis of the possible speedups that can be achieved by our approach, as a function of the diffusion steps T.

First, note that the expected number of tokens per draft can be derived from α and γ:

$$\mathbb{E}(\#\textit{tokens}) = \frac{1-\alpha^{\gamma+1}}{1-\alpha}. \tag{1}$$

In prior studies, theoretical results have focused on determining the optimal γ to maximize the throughput of speculative decoding methods [Leviathan et al., 2023]. For SpecDiff, extending γ introduces minimal overhead and becomes less important to consider. The number of sequential diffusion operations, T, instead impacts α as increasing this number improves the convergence of $M_q(x)$ to the learned distribution, aims to closely approximate the distribution of $M_p(x)$. Hence, an implicit dependency arises between T and α, which is reflected in Figure 2 (right).

First, note that the computation overhead of a single network evaluation of $M_q(x)$ and $M_p(x)$ is constant. $M_q(x)$ is scaled by the number of diffusion steps, whereas all evaluations of $M_p(x)$ are conducted in parallel. Now, consider that, provided Equation 1:

$$\mathbb{E}(\#\textit{tokens/second}) = \frac{\left(\frac{1-\alpha^{\gamma+1}}{1-\alpha}\right)}{Tc_1 + c_2} \tag{2}$$

where c_1 is the computation overheads of a single network evaluation of $M_q(x)$ and c_2 is the computation overheads of a single network evaluation of $M_p(x)$.

Next, consider that the convergence of $q(x)$ to the original data distribution, which we will denote as $\hat{q}(x)$, is proportional to the $1/T$.

Theorem B.1 (Li et al. [2023])**.** *Under standard assumptions, the convergence rate of samplers based on the probability flow Ordinary Differential Equation (ODE), converge at the rate*

$$TV(q_1, \hat{q}_1) \leq c_3 \frac{d^2 \log^4 T}{T} + c_3 \frac{d^6 \log^6 T}{T^2}$$
$$+ c_3 \sqrt{d \log^3 T \epsilon_{\textit{score}}} + c_3 d (\log T) \epsilon_{\textit{Jacobi}}$$

where d is the dimensions of the sample, $\epsilon_{\textit{score}}$ is the error in the score function estimation, $\epsilon_{\textit{Jacobi}}$ is the error in the Jacobian matrices, and universal constant $c_3 > 0$.

Given that $q_1 \approx q_0$, this result provides a practical upper bound on the distance between $q(x)$ and $\hat{q}(x)$. As Theorem B.1 provides an explicit relation to T, this can be used to determine an upper bound on the distance between $q(x)$ and $p(x)$ for a given number of diffusion steps. By the triangle inequality:

$$TV(p(x), q(x)) \leq TV(p(x), \hat{q}(x)) + TV(\hat{q}(x), q(x)) \tag{3}$$

Now, the remaining step to find the upper bound on the distance from $q(x)$ to $p(x)$ is to determine $TV(p(x), \hat{q}(x))$. First, consider that this relation can be expressed as follows:

Definition B.2 (Leviathan et al. [2023])**.** $D_{LK}(p,q) = \sum_x \|p(x) - M(x)\| = \sum_x \|q(x) - M(x)\|$ where $M(x) = \frac{p(x)+q(x)}{2}$

Corollary B.3 (Leviathan et al. [2023])**.** $\alpha = 1 - \mathbb{E}(D_{LK}(p,q)) = \mathbb{E}(\min(p,q))$

Provided Corollary B.3, $D_{LK}(p(x), \hat{q}(x))$ can be computed empirically by setting T arbitrarily high and evaluating α; subsequently, we will refer to this distance as c_4. We note that the $D_{LK}(p(x), \hat{q}(x))$ captures any error in the draft model's learned distribution that is introduced in Theorem B.1 as ϵ_{score} and ϵ_{Jacobi}, so we will set these to zero when applying the theorem.

Note that the metric D_{LK} is equivalent to the discretized total variation:

Proof.

$$\begin{aligned}D_{LK}(p,q) =& \sum_x \|p(x) - M(x)\| \\ =& \frac{1}{2}\sum_x \|p(x) - q(x)\| \\ \approx& \frac{1}{2}\int \|p(x) - q(x)\| dx \\ =& TV(p,q)\end{aligned}$$

□

For practical applications such as this, the discrete total variation is used, and for the purpose of this analysis we will consider the metrics equivalent. Now, we are ready to compute a lower bound for α that is dependent on T, applying Equation 3:

$$\begin{aligned}\alpha &= 1 - \mathbb{E}(D_{LK}(p,q)) = 1 - \mathbb{E}(TV(p,q)) \\ &\geq 1 - (c_4 + TV(\hat{q}(x), q(x)))\end{aligned}$$

$$\alpha \geq 1 - (c_4 + c_3\frac{d^2 \log^4 T}{T} + c_3\frac{d^6 \log^6 T}{T^2}) \tag{4}$$

Equation 4 provides a lower bound on α that is dependant on T. While in practice this remains computationally intractable, given that c_3 is unknown, this can be approximated using a surrogate network to predict this constant; this is not dissimilar from the suggestion by Leviathan et al. to optimize runtime using such an approach to predict α.

We are now ready to connect this to Equation 2:

$$\mathbb{E}\left(\#\textit{tokens/second}\right) \geq \frac{1-\left(1-c_4-c_3\frac{d^2\log^4 T}{T}-c_3\frac{d^6\log^6 T}{T^2}\right)^{\gamma+1}}{c_4+c_3\frac{d^2\log^4 T}{T}+c_3\frac{d^6\log^6 T}{T^2}} \times \frac{1}{Tc_1+c_2}$$

This equation can now be solved analytically to optimize the lower bound. Practically, in the presence of a surrogate network, this can be simplified further, given convergence of the diffusion model is proportional to $1/T$.

$$\mathbb{E}(\#\textit{tokens/second}) \geq \frac{1-(1-c_4-c_5\frac{1}{T})^{\gamma+1}}{c_4+c_5\frac{1}{T}} \times \frac{1}{Tc_1+c_2} \tag{5}$$

Hence, the dependency between T and α can be exploited to estimate the optimal number of diffusion steps. In our experiments we find that the optimal value of $T \leq 5$.

C Accepted Draft Lengths

We notice a starch contrast in the lengths of accepted drafts between these two settings. While the distribution of accepted draft lengths in Figure 3 (right) is what would be anticipated given the lower values of α, the longer generations on OpenWebText (left and middle of Figure 3) speak to the parallelism that can be realized when the distribution is effectively aligned to the target model.

D Implementation Details

For all evaluation we utilize the following hyperparameter setups. If not explicitly noted here, parameters are consistent with those specified by the authors of MDLM Sahoo et al. [2024]. For reported results on CNN/DM and OWT, 200 iterations are conducted on samples randomly selected from the datasets. For evaluation on MT bench, 480 iterations are conducted.

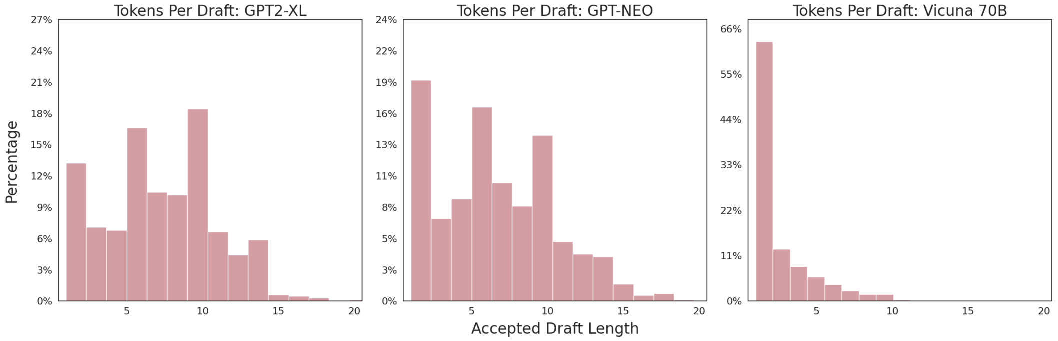

Figure 3: Accepted draft lengths for OpenWebText evaluation (left and middle) and MT Bench evaluation (right).

		M_p	M_q	Strategy	γ	T	Gen Length	Precision
CNN	**Ours**	GPT-2 XL (1.5B)	MDLM (110M)	*ddpm cache*	15	2	1024	32 bit
		GPT NEO (2.7B)	MDLM (110M)	*ddpm cache*	15	2	1024	32 bit
OWT	**Ours**	GPT-2 XL (1.5B)	MDLM (110M)	*ddpm cache*	15	2	1024	32 bit
		GPT NEO (2.7B)	MDLM (110M)	*ddpm cache*	20	2	1024	32 bit
MT	**Ours**	Vicuna (33B)	MDLM (141M)	*ddpm cache*	15	2	512	16 bit

Table 4: Additional details on parametric setups for reported results.

E Choice of Diffusion Model

The selection of a discrete diffusion model as the drafter plays a critical role in optimizing the overall framework's speedup performance. The models explored, MDLM and SEDD, represent the current state-of-the-art in discrete diffusion, achieving near auto-regressive perplexity results with comparably sized models. We observe significant speedups over SpS when using MDLM as our drafter, as MDLM demonstrates superior generation speeds overall. These gains are not observed in SEDD for two primary reasons: first, SEDD exhibits lower perplexity compared to MDLM, resulting in a lower acceptance rate and second, MDLM's generation speed surpasses that of SEDD.

		M_p	M_q	γ	α	Speedup
OWT	**Ours**	GPT-2 XL	SEDD Small	10	0.70	2.13x
		GPT NEO	SEDD Small	10	0.77	2.96x

Table 5: Performance of SpecDiff utilizing SEDD as the drafter. Note that unlike experiments in Tables 1 and 6, the draft model has been finetuned on the selected datasets; without finetuning, acceptance rates are below $\alpha = 0.2$, making SEDD impractical as a solely pretrained drafter.

F Temperature Impact on Acceptance Rate

As discussed in Section 7, overconfidence exhibited by discrete diffusion models, a result of poor model calibration, results in the top-1 token having a probability close to 1 for the vast majority of generations. For these outputs, the temperature is effectively zero. In the speculative decoding framework, this overconfidence is reflected as higher $q(x)$ values, which has a significant impact on the token acceptance rate when sampling stochastically. Specifically, high $q(x)$ values lead to a decrease in the number of accepted tokens because it increases the frequency of $q(x) > p(x)$. Increasing the temperature has a smoothing effect on $p(x)$ leading to misaligned distributions between $p(x)$ and $q(x)$. Consequently, tokens are more frequently rejected with a probability of $1 - \frac{p(x)}{q(x)}$, where a larger $q(x)$ further increasing the likelihood of rejections.

This obstacle is avoided when sampling deterministically (temperature=0) as in the results reported. This challenge motivates future study of discrete diffusion model calibration.

G Ablation Study: SpecDiff Without Initialization

While we observe a slight decrease in SpecDiff's speed when standard speculative decoding is not used to initialize the generation, the change in α is most noticeable when comparing to Table 1. This reflects the poor acceptance rates at the beginning of the generation. While SpecDiff does outperform SpS in this setting, *providing up to 1.5x speedups over SpS,* this highlights the benefit of adjointly using SpecDiff and other speculative decoding methods. Such hybrid approaches can be particularly effective for shorter generations.

		M_p	M_q	γ	α	Speedup
CNN	**Ours**	GPT-2 XL	MDLM	10	0.75	3.82x
		GPT NEO	MDLM	15	0.77	5.32x
OWT	**Ours**	GPT-2 XL	MDLM	15	0.78	4.06x
		GPT NEO	MDLM	15	0.82	6.19x

Table 6: Performance of a "vanilla" version of SpecDiff, that does not use SpS initialization.