Unlocking Efficiency in Large Language Model Inference: A Comprehensive Survey of Speculative Decoding

Heming Xia[1], **Zhe Yang**[2], **Qingxiu Dong**[2], **Peiyi Wang**[2],
Yongqi Li[1], **Tao Ge**[3], **Tianyu Liu**[4], **Wenjie Li**[1], **Zhifang Sui**[2]
[1]Department of Computing, The Hong Kong Polytechnic University
[2]National Key Laboratory for Multimedia Information Processing, Peking University
[3]Microsoft Research Asia [4]Alibaba Group
{he-ming.xia}@connect.polyu.hk; {yz_young}@pku.edu.cn

Abstract

To mitigate the high inference latency stemming from autoregressive decoding in Large Language Models (LLMs), Speculative Decoding has emerged as a novel decoding paradigm for LLM inference. In each decoding step, this method first efficiently drafts several future tokens and then verifies them in parallel. Unlike autoregressive decoding, Speculative Decoding facilitates the simultaneous decoding of multiple tokens per step, thereby accelerating inference. This paper presents a comprehensive overview and analysis of this promising decoding paradigm. We begin by providing a formal definition and formulation of Speculative Decoding. Then, we organize in-depth discussions on its key facets, including current leading techniques, the challenges faced, and potential future directions in this field. We aim for this work to serve as a catalyst for further research on Speculative Decoding, ultimately contributing to more efficient LLM inference.[1]

1 Introduction

Large Language Models (LLMs) have achieved remarkable proficiency in a range of downstream tasks (OpenAI, 2023; Touvron et al., 2023a,b; Chiang et al., 2023; Jiang et al., 2023). They are progressively evolving as the cornerstone of comprehensive API interfaces (e.g., ChatGPT[2]), offering human life services and guidance. However, the inference latency of these sizable models has emerged as a substantial obstacle restricting their further applications. This latency primarily arises from the token-by-token generation necessitated by autoregressive decoding, resulting in an escalation of the inference latency with both the length of the generated sequence and the model's scale.

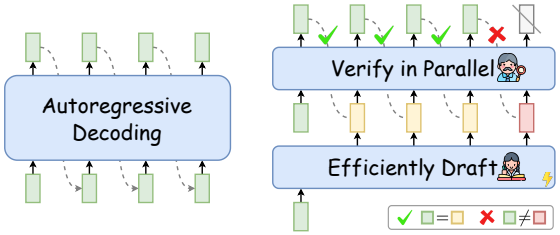

Figure 1: In contrast to autoregressive decoding (*left*) that generates sequentially, Speculative Decoding (*right*) first *efficiently drafts* multiple tokens and then *verifies* them *in parallel* using the target LLM. Drafted tokens after the bifurcation position (*e.g.,*) will be discarded to guarantee the generation quality.

To accelerate LLM inference, an innovative inference paradigm, Speculative Decoding, has been introduced (Stern et al., 2018; Xia et al., 2022; Leviathan et al., 2023; Chen et al., 2023a; Miao et al., 2023). As demonstrated in Figure 1, Speculative Decoding first leverages a drafter model to efficiently decode multiple tokens as speculation of future decoding steps and then uses the target LLM to verify the drafted tokens in parallel. Only those tokens that meet the LLM's verification criterion are accepted to guarantee high-quality outputs.

Speculative Decoding is founded upon two key observations about LLM inference: 1) many easy tokens can be predicted with less computation (e.g., using a smaller model), and 2) LLM inference is highly memory bandwidth bound (Patterson, 2004) with the main latency bottleneck arising from memory reads/writes of LLM parameters rather than arithmetic computations. Drawing on these observations, Speculative Decoding adapts the concept of *speculative execution*[3] to focus the LLM's computational efforts on the validation of multiple

[1]For ongoing reference, the relevant papers are summarized and will be regularly updated at https://github.com/hemingkx/SpeculativeDecodingPapers.

[2]https://chat.openai.com

[3]Speculative execution (Burton, 1985; Hennessy and Patterson, 2012) is an optimization technique used in computer architecture where tasks are performed in advance and subsequently verified for their necessity, thereby circumventing the delays inherent in sequential task execution.

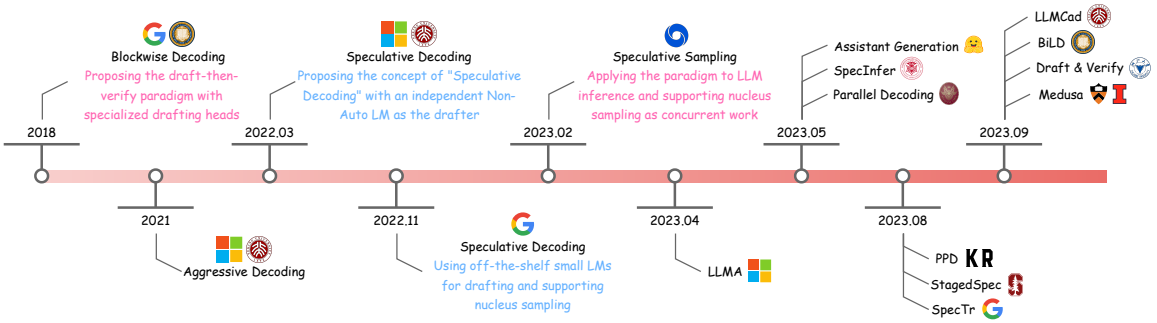

Figure 2: Timeline illustrating the evolution of Speculative Decoding. After 2022, Speculative Decoding was formally introduced as a general decoding paradigm to accelerate LLM inference and garnered widespread attention.

pre-drafted tokens, substantially diminishing the need for frequent memory reads/writes operations of LLM parameters, thereby improving inference efficiency.

While Speculative Decoding shows promise, it raises several critical questions that warrant further investigation. For instance, how to select or design the drafter model to strike a balance between speculation accuracy and drafting efficiency (Xia et al., 2022; Chen et al., 2023a; Santilli et al., 2023). Additionally, it is essential to examine whether the verification criterion can maintain both generation diversity and output quality (Miao et al., 2023; Spector and Re, 2023). Furthermore, careful consideration should be given to closely align the prediction behavior between the drafter and the target LLM for higher speculation accuracy (Zhou et al., 2023; Liu et al., 2023).

Amid the rapid expansion of research in Speculative Decoding, this work makes the first attempt to present a survey of this field, aiming to raise awareness within the academic community regarding the latest advancements. We provide a systematically organized categorization of current research and an in-depth analysis of relevant studies. Besides, we highlight the challenges and potential directions, hoping to serve as an essential guide for newcomers and to shed light on future research.

2 Overview

This paper offers a comprehensive survey of Speculative Decoding as a promising decoding paradigm for accelerating LLM inference. We commence by delivering an in-depth introduction to the early stages of Speculative Decoding research (§3), illustrated by a timeline of its evolution (as shown in Figure 2). This is followed by a formal definition and formulation of Speculative Decoding (§4). We present a taxonomy-based organizational framework to categorize relevant studies, as depicted in Figure 3. Then, this paper delves into a detailed discussion of leading techniques in Speculative Decoding, including the selection of drafter models (§5), verification strategies (§6), and alignment between the drafter and the target LLM (§7). Furthermore, we summarize several application scenarios where Speculative Decoding exhibits extraordinary effectiveness (§8). Finally, to facilitate beginners interested in this field, we outline the challenges faced and highlight potential directions for future research (§9).

3 Evolution of Speculative Decoding

3.1 Motivation

The widespread adoption of LLMs has established autoregressive decoding as the *de facto* standard to LLM inference (Chowdhery et al., 2023; OpenAI, 2023; Jiang et al., 2024). However, autoregressive decoding is limited by its inference latency, which primarily stems from the memory-bound computation of LLMs (Patterson, 2004; Shazeer, 2019). Specifically, the main latency bottleneck of each decoding step is not due to computational operations but arises from the necessity to transfer all LLM parameters from High-Bandwidth Memory (HBM) to the on-chip cache of modern accelerators like GPUs. This process, which generates only one token per step, leads to the underutilization of these accelerators and results in inefficiencies.

3.2 Pioneering *Draft-then-Verify* Efforts

To mitigate the above issue, an intuitive way is to trade off additional idle computational resources for more parallelism in LLM inference. To this end, Stern et al. (2018) introduced Blockwise Decoding, an approach that incorporates extra feedforward neural (FFN) heads atop the Transformer decoder, enabling the simultaneously *drafting* of multiple tokens per step. These tokens are then *verified* by the original LLM *in parallel*, ensuring that the outputs align with those of the original LLM. As a

Algorithm 1 Autoregressive Decoding

Require: Language model $\mathcal{M}_q$, input sequence $x_1, \ldots, x_t$, and target sequence length T;
initialize $n \leftarrow t$
while $n < T$ **do**
  Set $q_{n+1} \leftarrow \mathcal{M}_q\left(x \mid x_{<n+1}\right)$
  Sample $x_{n+1} \sim q_{n+1}$
  $n \leftarrow n+1$
end while

pioneering work proposing the *Draft-then-Verify* paradigm, Blockwise Decoding effectively reduces the total number of required LLM calls by increasing generation parallelism per step, thereby accelerating inference. However, the limited capacity of those extra FFN heads resulted in suboptimal drafting quality, leading to the underestimation of this paradigm.

To further unleash the potential of this promising paradigm, Xia et al. (2022) introduced Speculative Decoding (SpecDec), which utilizes an independent drafter, notably a specialized Non-Autoregressive Transformer, to perform the drafting task both accurately and efficiently. Besides, it presented an innovative verification strategy that relaxes the conventional rigid verification criterion, further increasing the acceptance rate of drafted tokens. Impressively, SpecDec achieves around $5\times$ speedup over conventional autoregressive decoding with comparable generation quality, underscoring the substantial potential of Speculative Decoding. Besides, this work marks the first time that the idea of speculative execution (Burton, 1985) is explicitly exploited for LLM acceleration.

Following SpecDec, Leviathan et al. (2023) and Chen et al. (2023a) made concurrent contributions by proposing Speculative Sampling, expanding this paradigm to encompass the lossless acceleration of nucleus sampling. These methods employed smaller LMs from the same series (e.g., `T5-small`) to speed up the inference of their larger counterparts (e.g., `T5-XXL`). Compared to previous work, these *off-the-shelf* small LMs do not require additional training, enabling the rapid adoption of Speculative Decoding in LLM acceleration. This advancement has elevated Speculative Decoding to the forefront of LLM efficiency research, attracting widespread interest within the NLP community.

To sum up, these pioneering efforts in Speculative Decoding have gradually solidified the *Draft-then-Verify* paradigm, showcasing its promising potential in LLM acceleration. We provide a detailed categorization and discussion of these studies and subsequent research in the following sections.

Algorithm 2 Speculative Decoding

Require: Target language model $\mathcal{M}_q$, drafter model $\mathcal{M}_p$, input sequence $x_1, \ldots, x_t$, block size K, target sequence length T, drafting strategy DRAFT, verification criterion VERIFY, and correction strategy CORRECT;
initialize $n \leftarrow t$
while $n < T$ **do**
  // Drafting: obtain distributions from $\mathcal{M}_p$ efficiently
  Set $p_1, \ldots, p_K \leftarrow \textsc{Draft}\left(x_{\leq n}, \mathcal{M}_p\right)$
  // Drafting: sample K drafted tokens
  Sample $\widetilde{x}_i \sim p_i, i = 1, \ldots, K$
  // Verification: compute $K+1$ distributions in parallel
  Set $q_i \leftarrow \mathcal{M}_q\left(x \mid x_{\leq n}, \widetilde{x}_{<i}\right), i = 1, \ldots, K+1$
  // Verification: verify each drafted token
  for $i = 1 : K$ **do**
    if $\textsc{Verify}\left(\widetilde{x}_i, p_i, q_i\right)$ **then**
      Set $x_{n+i} \leftarrow \widetilde{x}_i$ and $n \leftarrow n+1$
    else
      $x_{n+i} \leftarrow \textsc{Correct}\left(p_i, q_i\right)$
      and Exit for loop.
    end if
  end for
  If all drafted tokens are accepted, sample next token $x_{n+1} \sim q_{K+1}$ and set $n \leftarrow n+1$.
end while

4 Formulation and Definition

In this section, we first provide a succinct overview of standard autoregressive decoding (§4.1). Then, we offer an in-depth exposition of Speculative Decoding (§4.2), which encompasses a formal definition, a comprehensive description of the methodology, and a detailed elaboration of the algorithm.

4.1 Autoregressive Decoding

Transformer-based LLMs typically make generations in an autoregressive manner. Given an input sequence $x_1, \ldots, x_t$, an autoregressive language model $\mathcal{M}_q$ generates the next token according to:

$$x_{t+1} \sim q_{n+1} = \mathcal{M}_q\left(x \mid x_{<t+1}\right), \qquad (1)$$

where q is the conditional probability distribution calculated by $\mathcal{M}_q$ and x_{t+1} denotes the next token sampled from q_{n+1}. We illustrate a detailed process in Algorithm 1.

As discussed in Section 3, although the standard autoregressive decoding offers desirable generation quality, it is strongly bound by memory bandwidth, resulting in low utilization of contemporary accelerator hardware. In this process, each memory-bound LLM call (i.e., an LLM forward step) produces merely a single token for the entire sequence, making the whole generation inefficient and time-consuming.

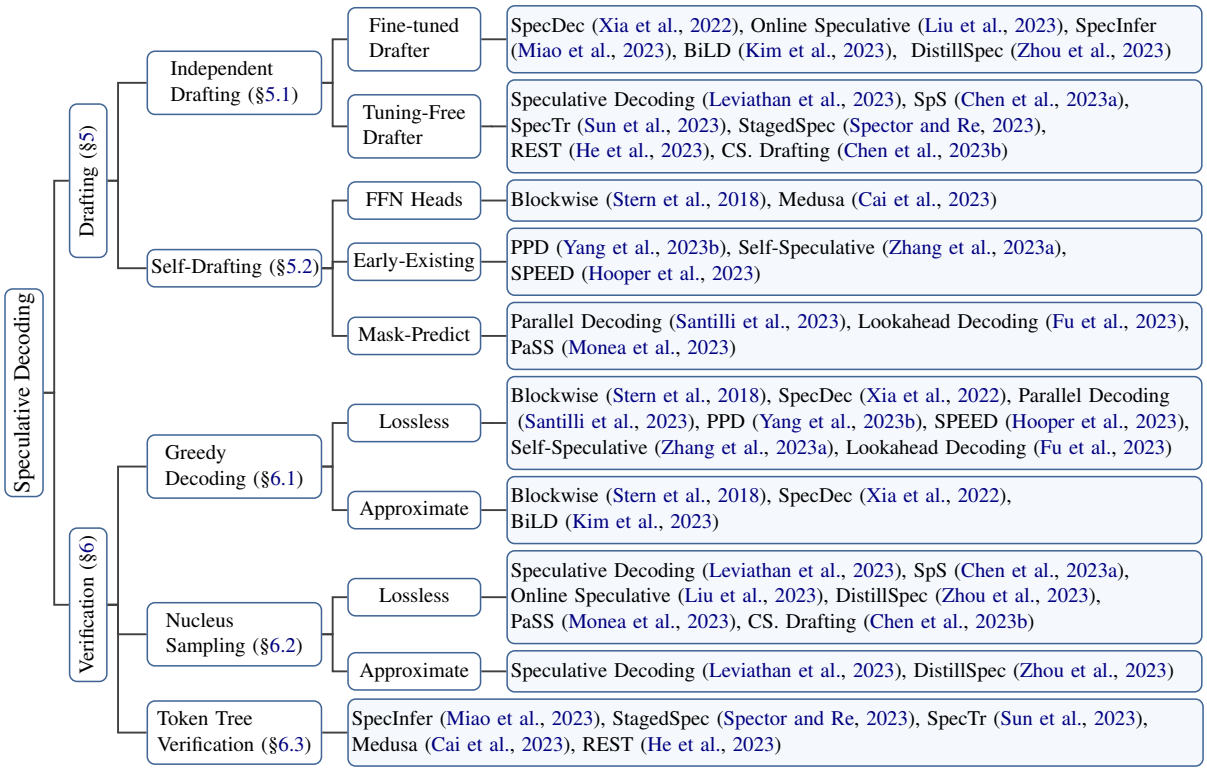

Figure 3: Taxonomy of Speculative Decoding.

4.2 Speculative Decoding

Following Xia et al. (2022), Leviathan et al. (2023), and Chen et al. (2023a), we here provide a formal definition of Speculative Decoding:

> Speculative Decoding is a *Draft-then-Verify* decoding paradigm in which, at each decoding step, it first *efficiently drafts* multiple future tokens and then *verifies* all these tokens *in parallel* using the target LLM to speed up inference.

We formulate a detailed Speculative Decoding process in Algorithm 2. Subsequently, we delve into the two fundamental substeps integral to this paradigm – *drafting* and *verification*:

Drafting At each decoding step, Speculative Decoding first efficiently drafts multiple future tokens, as a speculation of the target LLM's output. Formally, given an input sequence $x_1, \ldots, x_t$ and the target LLM $\mathcal{M}_q$, Speculative Decoding employs an efficient drafter model $\mathcal{M}_p$ (e.g., a smaller LM) to decode the next K drafted tokens:

$$\begin{aligned} p_1, \ldots, p_K &= \textsc{Draft}\left(x_{\leq t}, \mathcal{M}_p\right), \\ \widetilde{x}_i &\sim p_i, \quad i = 1, \ldots, K, \end{aligned} \tag{2}$$

where $\textsc{Draft}(\cdot)$ denotes various drafting strategies that we will discuss in Section 5, p is the conditional probability distribution calculated by $\mathcal{M}_p$, and $\widetilde{x}_i$ denotes the drafted token sampled from p_i.

Verification Subsequently, these drafted tokens are verified by the target LLM $\mathcal{M}_q$ in parallel. Formally, given the input sequence $x_1, \ldots, x_t$ and the draft $\widetilde{x}_1, \ldots, \widetilde{x}_K$, Speculative Decoding utilizes $\mathcal{M}_q$ to compute $K+1$ probability distributions simultaneously:

$$q_i = \mathcal{M}_q\left(x \mid x_{\leq t}, \widetilde{x}_{<i}\right), i = 1, \ldots, K+1. \tag{3}$$

Then, each drafted token $\widetilde{x}_i$ is verified by specific criterion – $\textsc{Verify}\left(\widetilde{x}_i, p_i, q_i\right)$. Only those tokens that meet the criterion are selected as final outputs, ensuring quality consistent with the target LLM's standards. Otherwise, the first drafted token $\widetilde{x}_c$ that fails the verification will be corrected by the strategy $\textsc{Correct}\left(p_c, q_c\right)$. All drafted tokens after position c will be discarded, to guarantee the high quality of the final outputs. If all tokens pass verification, an additional token x_{t+K+1} will be sampled from q_{K+1} as Eq. (1).

The drafting and verification substeps will be iterated until the termination condition is met, i.e., the `[EOS]` token is decoded or the sentence reaches the maximal length.

Consequently, the acceleration effect of Speculative Decoding primarily hinges on the number of

drafted tokens accepted per step. This acceptance rate is contingent on several factors, including the capacity of the drafter model, the verification criterion, and the behavior alignment between the drafter and the target LLM. Additionally, the intrinsic efficiency of the drafter itself also contributes to the overall end-to-end speedup. The subsequent section will delve into these pivotal components of Speculative Decoding, as depicted in Figure 3, through a comparative analysis of current leading methods.

5  Drafting

As a vital component of Speculative Decoding, the drafting process has a crucial impact on the acceleration effect of the paradigm. The impact is determined by two key factors: the speculation accuracy of the drafter $\mathcal{M}_p$, measured by the average number of accepted tokens per step, and the drafting latency (Stern et al., 2018; Xia et al., 2022). How to trade off high speculation accuracy and low drafting latency presents a major challenge in this process. In this section, we classify various drafting strategies $\textsc{Draft}\left(x_{\leq t}, \mathcal{M}_p\right)$ into two categories: independent drafting (§5.1) and self-drafting (§5.2), and summarize their formulations in Table 1.

5.1 Independent Drafting

To strike a balance between speculation accuracy and efficiency, SpecDec (Xia et al., 2022) first proposed to utilize an independent model to perform the drafting task. Specifically, it introduced a specialized Non-Autoregressive Transformer that drafts k tokens simultaneously per step. This model has a deep-shallow encoder-decoder architecture to run efficiently. Besides, SpecDec incorporated sequence-level knowledge distillation (Kim and Rush, 2016) to align the drafter's outputs with those of the target LLM, thereby improving speculation accuracy. However, this method requires training a specialized drafter model from scratch, which demands an increased computational budget.

Considering the available models in existing LLM series (e.g., OPT (Zhang et al., 2022) and LLaMA (Touvron et al., 2023a,b)), a more straightforward and efficient approach is directly employing a small LM from the same series as the drafter to accelerate the inference of its larger counterparts (Leviathan et al., 2023; Chen et al., 2023a; Spector and Re, 2023; Sun et al., 2023; Chen et al., 2023b). For instance, Leviathan et al. (2023) utilized `T5-small` as the drafter, to accelerate the inference of `T5-XXL`. These *off-the-shelf* small LMs do not require additional training or any modification on model architectures, facilitating the quick adoption of Speculative Decoding. Moreover, since models in the same series share tokenizers, pretraining corpora, and similar training processes, they inherently have an alignment in generation behaviors. Nevertheless, there is still a considerable behavior gap between the small LM and the target LLM, resulting in suboptimal speculation accuracy.

To improve behavior alignment, recent studies have investigated various knowledge distillation strategies to finetune existing small LMs as effective drafters (Miao et al., 2023; Kim et al., 2023; Zhou et al., 2023; Liu et al., 2023). Notably, Miao et al. (2023) proposed a collective boost-tuning strategy to align various small LMs with the target LLM on distinct subsets of the training corpus. The aggregated output of these small LMs, which are generated in parallel, offers an enhanced speculative prediction of the target LLM's outputs. Online Speculative Decoding (Liu et al., 2023) proposed to continually align the drafter with the target LLM on the user query data stream. It introduced an online knowledge distillation strategy that dynamically adapts the drafter model to the evolving distribution of user queries on the fly, thereby improving the speculation accuracy of the drafter.

5.2 Self-Drafting

While leveraging an external drafter model in Speculative Decoding shows promising speedup, this approach requires additional effort to train or identify a suitable drafter model that aligns closely with the target LLM. It becomes more challenging when the target LLM lacks smaller counterparts, e.g. LLaMA-7B (Touvron et al., 2023a,b). Moreover, the integration of two distinct models within one system introduces increased computational and operational complexities, especially in distributed settings (Cai et al., 2023).

To address the above issues, some work has proposed to utilize the target LLM itself for efficient drafting (Stern et al., 2018; Santilli et al., 2023; Hooper et al., 2023; Cai et al., 2023; Fu et al., 2023; Monea et al., 2023). Particularly, Blockwise Decoding (Stern et al., 2018) and Medusa (Cai et al., 2023) introduced additional FFN heads on top of the Transformer decoder, enabling the generation of multiple tokens simultaneously per step. Compared with external drafters, these lightweight

Methods	DRAFT $(x_{\leq t}, \mathcal{M}_p)$	Drafter Type
Parallel Drafting	$p_1, \ldots, p_K = \mathcal{M}_p(x \mid x_{\leq t})$	FFN Heads (Stern et al., 2018; Cai et al., 2023), Non-Autoregressive LM (Xia et al., 2022), Mask-Predict (Santilli et al., 2023; Fu et al., 2023)
Autoregressive Drafting	$p_i = \mathcal{M}_p(x \mid x_{\leq t}, \widetilde{x}_{<i}), i = 1, \ldots, K$	Small LM (Leviathan et al., 2023; Chen et al., 2023a), Early Existing (Yang et al., 2023b; Hooper et al., 2023), Layer Skipping (Zhang et al., 2023a)

Table 1: Summary of formulations for various drafting strategies in Speculative Decoding. We categorize these methods into two distinct groups based on their formulations: *parallel drafting* and *autoregressive drafting*.

FFN heads reduce extra computational overhead and are friendly to distributed inference. There is also another line of research utilized *early existing* or *layer skipping* on the target LLM itself to perform the drafting task (Yang et al., 2023b; Zhang et al., 2023a; Hooper et al., 2023). For instance, Yang et al. (2023b) introduces additional subprocesses that exist early in the current decoding step to start drafting future tokens in advance. Similarly, Self-Speculative (Zhang et al., 2023a) proposed to adaptively skip several intermediate layers during inference to draft efficiently.

In contrast to prior work that focused on extending model architectures or altering the inference process, Santilli et al. (2023) introduced a simple drafting strategy that directly adds multiple `[PAD]` tokens to the end of the input prompt. The effectiveness of this method stems from the robustness of LLMs in handling noisy inputs. Specifically, LLMs may still be capable of predicting the next token even with several `[PAD]` tokens inserted in the prefix. However, this approach deviates from the autoregressive pretraining pattern of LLMs, leading to suboptimal drafting quality. To tackle this problem, Fu et al. (2023) proposed to reform these low-quality drafted tokens into multiple n-grams, which effectively improves the speculation accuracy; Monea et al. (2023) introduced multiple learnable `[LA]` tokens and finetuned these token embeddings on a small training dataset to enhance the parallel decoding performance.

6 Verification

In each decoding step, the drafted tokens are then *verified in parallel*, to ensure the output quality is highly consistent with the target LLM. This process also determines the number of accepted tokens per step, a vital factor impacting the speedup. In this section, we summarize various verification criteria VERIFY $(\widetilde{x}_i, p_i, q_i)$ (as shown in Table 2), encompassing those supporting greedy decoding (§6.1) and nucleus sampling (§6.2) in LLM inference. Besides, we introduce token tree verification (§6.3), an effective strategy to increase token acceptance per step.

6.1 Greedy Decoding

Early attempts at Speculative Decoding focused on the verification criterion that supports greedy decoding, which guarantees that the outputs are exactly the same as the greedy decoding results of the target LLM (Stern et al., 2019; Sun et al., 2021; Xia et al., 2022). Specifically, this criterion requires that only those drafted tokens matching the top-1 predictions of the target LLM could pass the verification. Formally, given the input sequence $x_1, \ldots, x_t$, the drafted tokens $\widetilde{x}_1, \ldots, \widetilde{x}_K$, and the computed probability distributions $p_1, \ldots, p_K$, $q_1, \ldots, q_K$ as obtained from Eq. (2) and (3), respectively, the verification criterion VERIFY $(\widetilde{x}_i, p_i, q_i)$ on the i_{th} drafted token is formulated as

$$\widetilde{x}_i = \arg\max q_i, \tag{4}$$

where $i = 1, \ldots, K$. The first position c that the drafted token $\widetilde{x}_c$ fails the verification denotes the *bifurcation* position. The output token at this position x_{t+c} will be corrected by the correction strategy CORRECT (p_c, q_c), which simply replaces the drafted token with the top-1 prediction here:

$$x_{t+c} \leftarrow \arg\max q_c. \tag{5}$$

The verification criterion of greedy decoding is straightforward and effective. Thus, multiple subsequent studies have adopted this criterion to demonstrate the efficacy of their methodologies (Santilli et al., 2023; Yang et al., 2023b; Hooper et al., 2023; Zhang et al., 2023a; Fu et al., 2023). Besides, this criterion has been prominently featured in numerous online demonstrations (Joao Gante, 2023; Cai et al., 2023; Fu et al., 2023), highlighting how the algorithm generates faster than greedy decoding while maintaining identical outputs. However, this approach is not without its limitations. The strict matching requirement of this criterion often results

Methods	VERIFY $(\widetilde{x}_i, p_i, q_i)$	CORRECT (p_c, q_c)	Representative Work
Greedy Decoding	$\widetilde{x}_i = \arg\max q_i$	$x_{t+c} \leftarrow \arg\max q_c$	Blockwise Decoding (Stern et al., 2018), SpecDec (Xia et al., 2022)
Nucleus Sampling	$r < \min\left(1, \frac{q_i(\widetilde{x}_i)}{p_i(\widetilde{x}_i)}\right), r \sim U[0,1]$	$x_{t+c} \sim \text{norm}(\max(0, q_c - p_c))$	Speculative Decoding (Leviathan et al., 2023), SpS (Chen et al., 2023a)

Table 2: Summary of formulations for various verification strategies in Speculative Decoding.

in the rejection of potentially suitable drafted tokens, simply because they differ from the top-1 predictions of the target LLM, thereby constraining the speedup of the paradigm.

To tackle this problem, multiple studies have proposed various approximate verification criteria (Stern et al., 2018; Xia et al., 2022; Kim et al., 2023). Compared with the lossless greedy decoding criterion above, these methods slightly relax the matching requirement to trust the drafts more, leading to higher acceptance of drafted tokens. For instance, SpecDec (Xia et al., 2022) only requires the drafted tokens to fall in top-k candidates of the target LLM with a tolerable log-likelihood gap away from the top-1 prediction; BiLD (Kim et al., 2023) proposed a rollback verification criterion that rejects drafted tokens if the number of consecutive mismatch tokens exceeds a fixed threshold.

6.2 Nucleus Sampling

Following Stern et al. (2019) and Xia et al. (2022), subsequent work extended Speculative Decoding to support nucleus sampling (Leviathan et al., 2023; Chen et al., 2023a), accelerating the target LLM's inference without changing its output distribution. Formally, given the initial sequence $x_1, \ldots, x_t$, the drafted tokens $\widetilde{x}_1, \ldots, \widetilde{x}_K$ and the computed distributions $p_1, \ldots, p_K$, $q_1, \ldots, q_K$, the verification criterion VERIFY $(\widetilde{x}_i, p_i, q_i)$ on the i_{th} drafted token is

$$r < \min\left(1, \frac{q_i(\widetilde{x}_i)}{p_i(\widetilde{x}_i)}\right), r \sim U[0,1], \quad (6)$$

where r denotes a random number drawn from a uniform distribution $U[0,1]$; $q_i(\widetilde{x}_i)$ and $p_i(\widetilde{x}_i)$ are the probability of $\widetilde{x}_i$ according to $\mathcal{M}_q$ and $\mathcal{M}_p$, respectively; and $i = 1, \ldots, K$. In other words, this criterion accepts the drafted token $\widetilde{x}_i$ if $q_i(\widetilde{x}_i) \geq p_i(\widetilde{x}_i)$, and in case $q_i(\widetilde{x}_i) < p_i(\widetilde{x}_i)$ it rejects the token with probability $1 - \frac{q_i(\widetilde{x}_i)}{p_i(\widetilde{x}_i)}$. The corresponding correction strategy CORRECT (p_c, q_c) resamples the output token at the bifurcation position c from an adjusted distribution:

$$x_{t+c} \sim \text{norm}(\max(0, q_c - p_c)). \quad (7)$$

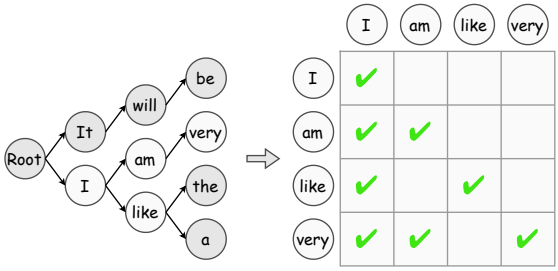

Figure 4: Illustration of the token tree sequences (*left*) and tree attention matrix (*right*). For simplicity, we only visualize the tree attention of tokens in white colors.

Leviathan et al. (2023) and Chen et al. (2023a) theoretically proved that Speculative Decoding with this sampling strategy maintains identical output distributions to the target LLM, which has been followed by multiple subsequent studies (Liu et al., 2023; Zhou et al., 2023; Monea et al., 2023; Chen et al., 2023b). In addition to the strict requirement in Eq. (6), some work also introduces approximate verification strategies to improve the rate of drafted token acceptance (Leviathan et al., 2023; Zhou et al., 2023). For instance, Leviathan et al. (2023) multiplies $p_i(\widetilde{x}_i)$ in Eq. (6) by a lenience parameter $l \in [0,1]$, relaxing the criterion of trusting the draft more.

6.3 Token Tree Verification

As illustrated in Table 2, prior verification strategies focus on verifying a single draft sequence and lack the consideration of different draft candidates, leading to suboptimal speculation accuracy. To address this issue, SpecInfer (Miao et al., 2023) first proposed token tree verification, an effective strategy that enables the target LLM to verify multiple candidate draft sequences in parallel.

As illustrated in Figure 4, compared to prior strategies that only verify a single draft sequence, token tree verification first merges multiple candidate draft sequences into a *token tree*, then utilizes a specially designed *tree attention* mechanism to verify the whole token tree in parallel. Recent research has investigated various approaches

	Methods	Drafting			Verification			Target LLM	Speedup (reported)
		Approach	Alignment	Tuning-free	Greedy	Nucleus	Token Tree		
Independ-D	SpecDec (Xia et al., 2022)	Non-Auto LM	Seq-KD	✗	✓	✗	✗	Transformer-base (65M)	3.9× ∼ 5.1×
	SpS (Chen et al., 2023a)	Small LM	-	✓	✓	✓	✗	Chinchilla (70B)	1.9× ∼ 2.5×
	BiLD (Kim et al., 2023)	Small LM	Seq-KD	✗	✗	✗	✗	T5-large (780M)	1.5× ∼ 2.1×
	SpecInfer (Miao et al., 2023)	Boost-tuned LMs	Col-BT	✗	✓	✓	✓	LLaMA (30B-65B)	2.0× ∼ 2.4×
	DistillSpec (Zhou et al., 2023)	Small LM	KD	✗	✓	✓	✗	T5-XL (3B)	-
	Online Speculative (Liu et al., 2023)	Small LM	Online-KD	✗	✓	✓	✗	Vicuna (7B)	-
	CS. Drafting (Chen et al., 2023b)	Cascaded LMs	-	✓	✓	✓	✗	FLAN-T5-xxl (11B)	-
	REST (He et al., 2023)	Context Retrieval	-	✓	✓	✓	✓	Vicuna (7B-13B)	1.7× ∼ 1.8×
Self-D	Blockwise Decoding (Stern et al., 2018)	FFN Heads	Seq-KD	✗	✓	✗	✗	Transformer-big (213M)	1.7× ∼ 3.0×
	Medusa (Cai et al., 2023)	FFN Heads	-	✗	✓	✗	✓	Vicuna (7B-33B)	1.9× ∼ 2.0×
	PPD (Yang et al., 2023b)	Early Existing	-	✗	✓	✗	✗	Vicuna (13B)	1.1× ∼ 1.5×
	Self-Speculative (Zhang et al., 2023a)	Layer Skipping	-	✓	✓	✗	✗	LLaMA-2 (13B-70B)	1.4× ∼ 1.5×
	Parallel Decoding (Santilli et al., 2023)	Mask-Predict	-	✓	✓	✗	✗	MBart50 (610M)	1.0× ∼ 1.1×
	Lookahead Decoding (Fu et al., 2023)	Mask-P & N-grams	-	✓	✓	✗	✗	LLaMA-2 (7B-70B)	1.5× ∼ 2.3×
	PaSS (Monea et al., 2023)	Learnable Tokens	-	✗	✓	✓	✗	LLaMA (7B)	1.3× ∼ 1.4×

Table 3: Summary of various Speculative Decoding methods. "*Independ-D*" and "*Self-D*" denote independent drafting and self-drafting, respectively. "*Greedy*", "*Nucleus*", and "*Token Tree*" denote whether the method supports greedy decoding, nucleus sampling, and token tree verification, respectively. We list the most representative target LLMs for each method and the speedups in the original paper (if reported), which is obtained with a batch size of 1.

to obtain the candidate draft sequences. For instance, Miao et al. (2023) generated diverse draft sequences from different boost-tuned LMs; Cai et al. (2023) considered the top-k predictions from each FFN head to obtain multiple candidates, while He et al. (2023) utilized different continuations (of the input prompt) from the retrieved documents as candidate draft sequences. Subsequently, those obtained candidate draft sequences are merged into a token tree by sharing prefixes and are fed into the target LLM with a tree attention mask for parallel verification, as shown in Figure 4.

7 Alignment

As illustrated in Section 5, improving speculation accuracy is the key to the speedup of Speculative Decoding: the closer the prediction behavior of the drafter is to the target LLM, the higher the acceptance rate of drafted tokens. To this end, existing work has explored various knowledge distillation (KD) strategies to align the drafter's outputs with those of the target LLM (Stern et al., 2018; Xia et al., 2022; Miao et al., 2023; Liu et al., 2023; Kim et al., 2023; Zhou et al., 2023). Blockwise Decoding first adopted sequence-level knowledge distillation (Seq-KD) (Kim and Rush, 2016) for alignment, which trained the drafter model on the sentences generated by the target LLM. Besides, Seq-KD is also an effective strategy to improve the generation quality of parallel decoding (Gu et al., 2018; Qian et al., 2021), which enhances the drafting performance. Thus, this approach has been adopted by multiple subsequent studies (Xia et al., 2022; Miao et al., 2023; Kim et al., 2023). For instance, Miao et al. (2023) proposed a collective boost-tuning (Col-BT) strategy, which adopted Seq-KD to finetune multiple small LMs on the training data and utilized their aggregated output as the draft, improving the speculation accuracy.

Though Seq-KD is effective, it ignores the probability distributions of the target LLM and only trains the drafter model with one-hot labels. Thus, this strategy becomes less effective when Speculative Decoding is adopted for nucleus sampling. To address this, recent studies have explored other KD strategies for Speculative Decoding (Zhou et al., 2023; Liu et al., 2023). Notably, DistillSpec (Zhou et al., 2023) conducted a comprehensive comparison of different KD strategies on Speculative Decoding across various downstream tasks, pointing out that the choice of the optimal KD algorithm largely depends on specific tasks and the verification strategy. Online Speculative (Liu et al., 2023) proposed an online KD strategy that dynamically aligns the drafter with the target LLM on the fly using the query data.

We summarize the main features of existing Speculative Decoding methods in Table 3, including the drafter type or the drafting strategy, the alignment approach, supported verification strategies, and the reported speedup, etc.

8 Applications

In addition to serving as a general paradigm, recent work has revealed that some variants of Speculative Decoding demonstrate extraordinary effectiveness in specific tasks. Furthermore, other research has applied this paradigm to address latency issues unique to certain application scenarios, achieving inference acceleration. Below, we will provide a

detailed introduction to these promising works.

Recent studies by Sun et al. (2021) and Yang et al. (2023a) have highlighted Speculative Decoding is particularly well suited for tasks where model inputs and outputs are highly similar, such as Grammatical Error Correction (Wang et al., 2021; Bryant et al., 2023) and Retrieval-augmented Generation (Lewis et al., 2020; Cai et al., 2022). These methods introduced a specialized form of Speculative Decoding, where the initial user input or the retrieved context is directly employed as drafts. For instance, SAD (Sun et al., 2021), an early attempt at Speculative Decoding on Grammatical Error Correction, utilized the input sentence with grammatical errors as a draft and leveraged the LLM to verify the whole sentence in parallel, achieving a $9\times\sim12\times$ speedup. Similarly, LLMA (Yang et al., 2023a) selected text spans from the reference as drafts, demonstrating a $2\times\sim3\times$ speedup across various practical application scenarios including Retrieval-augmented Generation, Cache-assisted Generation, and Multi-turn Conversations.

Beyond these works, RaLMSpec (Zhang et al., 2023b) adopted Speculative Decoding to accelerate retrieval-augmented language models (RaLMs). It pointed out that the main latency bottleneck of iterative RaLMs is the frequent retrieval from a vast knowledge base. To accelerate inference, this method proposed to maintain a local cache for speculative retrieval, achieving around $2\times$ speedup with identical model outputs. LLMCad (Xu et al., 2023) applied Speculative Decoding to on-device LLM inference. Concretely, it proposed to generate drafts with a smaller real-time LM that can be hosted in device memory, and only utilize the target LLM for parallel verification. This approach effectively reduces repetitive releasing and loading of model weights, achieving a $9.3\times$ speedup compared to existing inference engines.

9 Challenges and Future Directions

How to trade off speculation accuracy and drafting efficiency? As discussed in Sections 5, scaling up the drafter can effectively enhance speculation accuracy, yet it largely reduces the drafting efficiency and even the overall speedup. Therefore, it is essential to strike a balance between speculation accuracy and drafting latency. Among existing strategies, behavior alignment is a promising approach to address this issue, as it improves speculation accuracy without increasing latency. However, despite recent advancements (Miao et al., 2023; Zhou et al., 2023; Liu et al., 2023), there is still considerable room for improvement to align the drafter with the target LLM. For example, given that the drafted tokens after the bifurcation position are all discarded, one potential direction could involve encouraging the drafter to prioritize the generation quality of early-position tokens. Beyond alignment, other factors such as the quality of drafting (Fu et al., 2023) and the determination of speculation length (Su et al., 2023) also influence speculation accuracy and merit further exploration.

How to integrate Speculative Decoding with other leading techniques? As a general decoding paradigm, Speculative Decoding has already demonstrated its potential in conjunction with other advanced techniques (Yang et al., 2023a; Zhang et al., 2023b; Li et al., 2023). For instance, Yuan et al. (2023) combined Speculative Decoding with Contrastive Decoding (Li et al., 2023), which not only speeds up the inference but also substantially improves the generation quality. In addition to the acceleration of text-only LLMs, the application of Speculative Decoding in multimodal inference, such as image synthesis, text-to-speech synthesis, and video generation, is also an intriguing and valuable direction for future research.

10 Conclusion

In recent years, the continual scaling up of LLMs has significantly increased the demand for efficient LLM inference. Speculative Decoding, a novel decoding paradigm that accelerates LLM inference while maintaining identical generation quality, has emerged as a promising solution. This paper presents a comprehensive survey of the existing literature on Speculative Decoding, including a formal definition and formulation of Speculative Decoding, an in-depth review of various leading techniques, as well as challenges and potential directions for future research. To the best of our knowledge, this is the first survey dedicated to Speculative Decoding. The primary objective of this paper is to clarify the current research landscape and provide insights into the future trajectory of this promising paradigm.

References

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe.

2023. Grammatical error correction: A survey of the state of the art. Comput. Linguistics, 49(3):643–701.

F. Warren Burton. 1985. Speculative computation, parallelism, and functional programming. IEEE Trans. Computers, 34(12):1190–1193.

Deng Cai, Yan Wang, Lemao Liu, and Shuming Shi. 2022. Recent advances in retrieval-augmented text generation. In SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022, pages 3417–3419. ACM.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, and Tri Dao. 2023. Medusa: Simple framework for accelerating llm generation with multiple decoding heads. https://github.com/FasterDecoding/Medusa.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023a. Accelerating large language model decoding with speculative sampling. CoRR, abs/2302.01318.

Ziyi Chen, Xiaocong Yang, Jiacheng Lin, Chenkai Sun, Jie Huang, and Kevin Chen-Chuan Chang. 2023b. Cascade speculative drafting for even faster llm inference.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. J. Mach. Learn. Res., 24:240:1–240:113.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. 2023. Breaking the sequential dependency of llm inference using lookahead decoding.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In International Conference on Learning Representations.

Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D. Lee, and Di He. 2023. REST: retrieval-based speculative decoding. CoRR, abs/2311.08252.

John L. Hennessy and David A. Patterson. 2012. Computer Architecture - A Quantitative Approach, 5th Edition. Morgan Kaufmann.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Hasan Genc, Kurt Keutzer, Amir Gholami, and Yakun Sophia Shao. 2023. SPEED: speculative pipelined execution for efficient decoding. CoRR, abs/2310.12072.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. CoRR, abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Joao Gante. 2023. Assisted generation: a new direction toward low-latency text generation.

Sehoon Kim, Karttikeya Mangalam, Jitendra Malik, Michael W. Mahoney, Amir Gholami, and Kurt Keutzer. 2023. Speculative decoding with big little decoder. CoRR, abs/2302.07863.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. arXiv preprint arXiv:1606.07947.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of Proceedings of Machine Learning Research, pages 19274–19286. PMLR.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In

Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 12286–12312. Association for Computational Linguistics.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Ion Stoica, Zhijie Deng, Alvin Cheung, and Hao Zhang. 2023. Online speculative decoding. CoRR, abs/2310.07177.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Rae Ying Yee Wong, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. 2023. Specinfer: Accelerating generative LLM serving with speculative inference and token tree verification. CoRR, abs/2305.09781.

Giovanni Monea, Armand Joulin, and Edouard Grave. 2023. Pass: Parallel speculative sampling. CoRR, abs/2311.13581.

OpenAI. 2023. GPT-4 technical report. CoRR, abs/2303.08774.

David A. Patterson. 2004. Latency lags bandwith. Commun. ACM, 47(10):71–75.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 1993–2003. Association for Computational Linguistics.

Andrea Santilli, Silvio Severino, Emilian Postolache, Valentino Maiorca, Michele Mancusi, Riccardo Marin, and Emanuele Rodolà. 2023. Accelerating transformer inference for translation via parallel decoding. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023, pages 12336–12355. Association for Computational Linguistics.

Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need. CoRR, abs/1911.02150.

Benjamin Spector and Chris Re. 2023. Accelerating LLM inference with staged speculative decoding. CoRR, abs/2308.04623.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of Proceedings of Machine Learning Research, pages 5976–5985. PMLR.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018. Blockwise parallel decoding for deep autoregressive models. In Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pages 10107–10116.

Qidong Su, Christina Giannoula, and Gennady Pekhimenko. 2023. The synergy of speculative decoding and batching in serving large language models. CoRR, abs/2310.18813.

Xin Sun, Tao Ge, Furu Wei, and Houfeng Wang. 2021. Instantaneous grammatical error correction with shallow aggressive decoding. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pages 5937–5947. Association for Computational Linguistics.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix X. Yu. 2023. Spectr: Fast speculative decoding via optimal transport. CoRR, abs/2310.15141.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. CoRR, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan,

Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. CoRR, abs/2307.09288.

Yu Wang, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. A comprehensive survey of grammatical error correction. ACM Trans. Intell. Syst. Technol., 12(5):65:1–65:51.

Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. 2022. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation.

Daliang Xu, Wangsong Yin, Xin Jin, Ying Zhang, Shiyun Wei, Mengwei Xu, and Xuanzhe Liu. 2023. Llmcad: Fast and scalable on-device large language model inference. CoRR, abs/2309.04255.

Nan Yang, Tao Ge, Liang Wang, Binxing Jiao, Daxin Jiang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023a. Inference with reference: Lossless acceleration of large language models. arXiv preprint arXiv:2304.04487.

Seongjun Yang, Gibbeum Lee, Jaewoong Cho, Dimitris S. Papailiopoulos, and Kangwook Lee. 2023b. Predictive pipelined decoding: A compute-latency trade-off for exact LLM decoding. CoRR, abs/2307.05908.

Hongyi Yuan, Keming Lu, Fei Huang, Zheng Yuan, and Chang Zhou. 2023. Speculative contrastive decoding. CoRR, abs/2311.08981.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2023a. Draft & verify: Lossless large language model acceleration via self-speculative decoding. CoRR, abs/2309.08168.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. CoRR, abs/2205.01068.

Zhihao Zhang, Alan Zhu, Lijie Yang, Yihua Xu, Lanting Li, Phitchaya Mangpo Phothilimthana, and Zhihao Jia. 2023b. Accelerating retrieval-augmented language model serving with speculation. In Submitted to The Twelfth International Conference on Learning Representations. Under review.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. 2023. Distillspec: Improving speculative decoding via knowledge distillation.