Temperature-Centric Investigation of Speculative Decoding with Knowledge Distillation

Anonymous ACL submission

Abstract

Speculative decoding stands as a pivotal technique to expedite inference in autoregressive (large) language models. This method employs a smaller *draft* model to speculate a block of tokens, which the *target* model then evaluates for acceptance. Despite a wealth of studies aimed at increasing the efficiency of speculative decoding, the influence of generation configurations on the decoding process remains poorly understood, especially concerning decoding temperatures. This paper delves into the effects of decoding temperatures on speculative decoding's efficacy. Beginning with knowledge distillation (KD), we first highlight the challenge of decoding at higher temperatures, and demonstrate KD in a consistent temperature setting could be a remedy. We also investigate the effects of out-of-domain testing sets with out-of-range temperatures. Building upon these findings, we take an initial step to further the speedup for speculative decoding, particularly in a high-temperature generation setting. Our work offers new insights into how generation configurations drastically affect the performance of speculative decoding, and underscores the need for developing methods that focus on diverse decoding configurations.

1 Introduction

Large language models (LLMs) such as GPT-4 (OpenAI, 2023), Claude (Bai et al., 2022), and LLaMA (Touvron et al., 2023a,b) are revolutionizing the field of natural language processing (NLP) and machine learning (ML). While being powerful tools for various downstream tasks, LLMs' real-time deployment is still challenging due to the size and the inference cost (Pope et al., 2022). Conversely, smaller models have less latency but lower generative quality. In a word, efficiency and accuracy form a trade-off. Inspired by this, speculative decoding (Leviathan et al., 2023; Chen et al., 2023a) emerges as a promising *token-level* solution to reduce the latency of generation for LLMs. Specifically, speculative decoding leverages smaller models as draft models to speculate successive candidate tokens for multiple inference steps with autoregressive generation, which are then verified with the target LLM in parallel through a *single forward pass*. If a token fails to be accepted by the target LLM, all the consecutive tokens will be discarded, and the target LLM needs resampling for that rejected token.

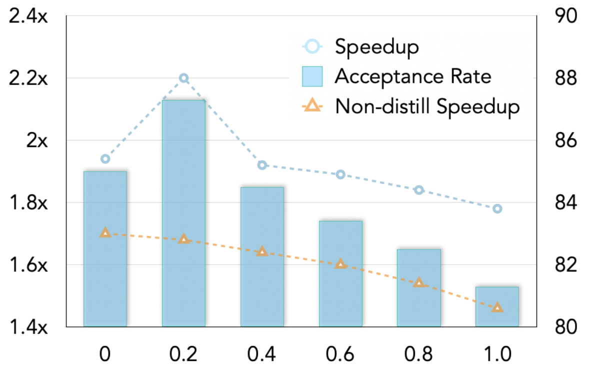

Figure 1: Speedup and acceptance rate (y-axises) for different decoding temperatures (x-axis) on Alpaca dataset. The draft model (Llama-68M) is distilled from Llama-2-13B-chat with data generated in 0.2 temperature.

Previous studies (Xia et al., 2024) generally test speculative decoding in fixed generation configurations, with temperature sampling (Ackley et al., 1985) being the default setting. Compared with other hyperparameters such as top-k (Fan et al., 2018) in text generation, temperature has a dominating effect in re-estimating the distribution before top-k sampling (Radford and Narasimhan, 2018), balancing generation quality and diversity (Holtzman et al., 2020). However, previous works only test at a coarse-grained level, setting the temperature to binary extremes of either 0 (greedy decoding) or 1.0. On the other hand, accelerating speculative decoding in various generation scenarios is important to better suit user needs in downstream tasks. To this end, this paper investigates from a **temperature-centric** perspective of speculative decoding for LLMs.

We focus on knowledge distillation (KD) (Hinton et al., 2015) as the general investigation setting, which has been introduced as an intuitive and general solution to speculative decoding (Zhou et al., 2023; Liu et al., 2023). Particularly, KD aims to align the distributions of draft models better to that of target models. In this way, the *acceptance rate* of candidate tokens generated by the draft model to the target model could be boosted. Our preliminary experiments in Figure 1 validate our motivation, highlighting the impacts brought by different temperatures for both decoding and KD stages. Notably, the speedup of the decoding processes increases and peaks at a decoding temperature of 0.2 before declining as the temperature approaches 1. The impact of temperature on speedup can reach a relative difference of around **30%** ($\frac{2.23-1.72}{1.72} = 29.7\%$), highlighting its importance. We also notice that KD relieves the degradation of speedup when temperature increases.

Overall, we explore the impact of temperature on speculative decoding with KD. Specifically, we address three pivotal research questions:

- **RQ1. What is the influence of temperature on speculative decoding's efficacy in the context of KD?** To answer this question, we explore two key processes where the temperature is a critical factor in speculative decoding (§ 2). Utilizing the Llama series as the foundational model for both target and draft models, we train the draft model across a spectrum of training sets, each regulated by nuanced temperature settings, to assess and benchmark their performance (§ 4.1).

- **RQ2. Can the observed results extrapolate to out-of-domain datasets and out-of-range temperatures?** Building upon RQ1, we examine the adaptability of KD with temperature-specific configurations to *out-of-domain* test sets derived from the training sets (§ 4.2), and its performance with *out-of-range* temperatures from those used during training (§ 4.3).

- **RQ3. How do we design an efficient recipe for enhancing speculative decoding in a temperature-centric manner?** Drawing from the insights of RQ1, we investigate various strategies for assembling training data with a temperature-aware approach (§ 4.4). Our goal is to amplify the performance of speculative decoding, particularly under conditions of elevated decoding temperatures.

The experiments are conducted on several commonly used public datasets. Our analysis offers a new perspective on understanding speculative decoding by applying fine-grained temperature controls, especially with KD. The key contributions and takeaways can be summarized as follows:

- We pinpoint *temperature* as the key factor in the process of speculative decoding with KD. We empirically identify the most suitable setup, and find that temperature alignment between training and inference accelerates decoding significantly.

- We explore both *out-of-domain* test sets and *out-of-range* decoding temperatures, and show the importance of token difficulties for out-of-domain sets and the "U-curve" phenomenon for out-of-range temperatures.

- Building upon our findings, we propose a simple yet effective data-centric strategy to boost the speedup for speculative decoding at high temperatures, and show that it can further improve the speedup of 12%-20%.

2 Background

2.1 Temperature in Decoding

Temperature τ is an important hyperparameter in the configurations for decoding, which controls the randomness of predictions by scaling the logits before applying the softmax function during the text generation process (Ackley et al., 1985). It affects how the next word is chosen from the vocabulary:

$$\mathbb{P}(t_k|t_{1:k-1}) = \frac{exp(l_k/\tau)}{\sum_i exp(l_i/\tau)}, \quad (1)$$

where t_k and l_k are the k-th token to predict and the corresponding logit. Lower temperatures will skew the distribution toward high-probability events, reducing the mass in the tail distribution to make the generation more focused and deterministic, and *vice versa*.

2.2 Temperature in Knowledge Distillation

The latency reduction actually depends on how *aligned* the draft model and the LLM are. With better alignment comes lower rejection rates of tokens, thus higher acceleration speed. To make draft models better aligned with LLMs, KD is proposed as an intuitive yet effective solution (Zhou et al., 2023; Liu et al., 2023). In the KD process, the draft model $\mathcal{M}_d$ acts as the student, and the target model $\mathcal{M}_t$ serves as the teacher. We consider the two KD paradigms, *online* and *offline* distillation (Zhong et al., 2024), in our investigation. Note that this paper focuses on lossless speculative decoding and the detailed algorithm for KD can be found in Appendix A.

During the KD process, the effect of temperature is mostly brought by the process of training data ($\mathcal{G}$) generation, which is contrastive to the temperature used in loss functions[1]. Temperature guides the training data generation from the teacher model for offline data inference. Similar to offline distillation, the student model is asked to generate on-policy training data with temperature being the controlling factor in online distillation (Agarwal et al., 2023).

Offline Distillation We use SeqKD (Kim and Rush, 2016) as the representative technique for offline distillation. It is a *black-box* style framework, where only the teacher-generated texts are accessible. Training data are first generated by teacher $\mathcal{M}_t$ with decoding temperature τ:

$$\begin{aligned} y_i &= \mathcal{M}_t(x_i; \tau) \\ \mathcal{G} &= \{(x_i, y_i) \mid i = 1, 2, ..., n\} \end{aligned} \tag{2}$$

where (x_i, y_i) denotes the input-output pair. The collected data are then used to train the student $\mathcal{M}_d^\theta$ parameterized by θ:

$$\theta^* = \arg\min_\theta \sum_{(x_i,y_i)\in\mathcal{G}} \mathcal{L}(\mathcal{M}_d^\theta(x_i), y_i) \tag{3}$$

The student model $\mathcal{M}_d^\theta$ is trained to minimize this loss, effectively learning to mimic the teacher's behavior.

Online Distillation In this setting, we assume *white-box* access to both target and draft models, i.e., we can obtain the token-level distributions. Online distillation to the draft models seeks to minimize the divergence between the soft logits of teacher and student distributions over a training set, by using online data generated by $\mathcal{M}_d$:

$$\theta := \arg\min \mathbb{E}_{(x,y)\sim\mathcal{G}}[D(\mathcal{M}_t || \mathcal{M}_d^\theta(y|x;\tau;\lambda))],$$

[1] In our investigation, the temperature in loss functions is always set to 1.0 following previous works (Chang et al., 2023).

Setting	Divergence ($\mathcal{D}$)	Training Data ($\mathcal{G}$)
Offline Distill	FKL	Data generated by $\mathcal{M}_t$ offline
Online Distill	FKL	Fixed dataset + Online data generated by $\mathcal{M}_d$

Table 1: Comparison of two settings for offline distillation and online distillation.

$\mathcal{D}$ measures the distance of two distributions and we use the default *forward* Kullback-Leibler divergence (FKL) in our experiments. τ and λ control the generation temperature and data fractions of the student model, respectively. Table 1 summarizes the setting of offline and online distillation.

3 Experimental Setup

This section outlines the detailed experimental setup, including model architecture, dataset selection, and evaluation metric employed for knowledge distillation (KD) and decoding phases. Further details on implementation, including hyperparameter configurations and computation timeframes, are provided in Appendix B.

3.1 Models and Datasets

Models In our experiments, we follow the settings of previous works (Liu et al., 2023; Miao et al., 2023) and employ the Llama (Touvron et al., 2023a,b) series as model architectures, a publicly available and prevalently used LLM family. Specifically, we select the instruction-tuned Llama-2-13B-chat [2] as the target model, and Llama-68M [3] as the draft model. The pre-trained model parameters for both models used are accessible via HuggingFace.

Datasets We focus on the general task of text generation with instructions. We use the Alpaca (Taori et al., 2023) dataset as our fixed dataset following (Miao et al., 2023). The original Alpaca collection contains 52k samples in the format of instruction-input-output triples, and we take $51k$ as the training set for KD. The rest of the $1k$ samples are left as *in-domain* testing set. For offline distillation, we employ vLLM (Kwon et al., 2023) first to generate responses for each sample in the fixed dataset using the teacher model $\mathcal{M}_t$. The generated responses are paired with the original input as the training data for offline distillation. For online distillation, we use a half-fixed dataset with another half-on-policy data generated by student

[2] https://huggingface.co/meta-llama/Llama-2-13b-chat-hf

[3] https://huggingface.co/JackFram/llama-68m

model $\mathcal{M}_d$. All data generated by either $\mathcal{M}_t$ or $\mathcal{M}_d$ is based on temperature sampling with temperature τ in $[0, 1]$ of interval 0.1. That being said, we have a total of 11 configurations of data generation in the KD process, which results in 22 draft models for testing for both offline and online distillation settings. Apart from the $1k$ samples from Alpaca as the *in-domain* set, we also use the GSM8K (Cobbe et al., 2021) test set containing $1.28k$[4] samples as the *out-of-domain* set.

3.2 Evaluation

Metrics Following previous works (Leviathan et al., 2023; Miao et al., 2023; Zhou et al., 2023), we measure the empirical acceptance rate α, and relative wall time (latency) improvement γ. α serves as the measure of how closely $\mathcal{M}_d$ approximates $\mathcal{M}_t$, and directly influences γ. In our implementations, we adapt the code from HuggingFace assisted decoding [5] and count the numbers of tokens generated by $\mathcal{M}_d$ and tokens accepted by $\mathcal{M}_t$ for α. Time for decoding is documented for γ.

All the decoding processes are conducted based on temperature sampling with temperature $\tau \in [0, 1]$ spanning 0.2. The batch size is set to 1 by default. For statistical robustness, we decode each sample 5 times and take the averaged number of α and γ and the final results.

Platforms The KD training was executed over eight V100 NVIDIA GPUs, each with 32GB memory. The decoding phase for all draft models was carried out on a single A100 40G NVIDIA GPU for the consistency of our conclusions.

4 Experiments and Analyses

Our experiments and analyses are organized in the following workflow. We start with an overall investigation of temperature configurations for two KD settings for in-domain testing. Leveraging these observations, we further test these insights on out-of-domain datasets with out-of-range temperatures. Finally, we brought out a simple yet effective solution to further improve the performance of speculative decoding with higher decoding temperatures.

[4]The original test set of GSM8K contains $1.32k$ samples, we filter out samples that exceed the context length of the draft model.

[5]https://huggingface.co/blog/assisted-generation

4.1 Overall Investigation

To quantify how temperature impacts the speculative decoding process, we plot the overall investigation results for both offline distillation and online distillation using 11 KD models trained with different temperatures under 6 decoding configurations in Figure 2 (a) and (b) respectively. We interpret the results in the following aspects. Additional analyses can be found in Appendix C.

Decoding at a high temperature is generally slower. First of all, we observe a consistent trend of diminishing speedup as the decoding temperature increased from 0 to 1. This trend corroborates the findings of previous studies, such as those by Xia et al. (2024). Our analysis revealed that this phenomenon persists across all KD temperatures, affecting both offline distillation and online distillation processes. The effect was most pronounced when the KD temperature was set to 0, leading to a relative speedup difference of 31% and 29% for offline distillation and online distillation, respectively. This is attributed to the increased computational complexity of the speculative sampling criterion at high temperatures, as demonstrated in prior research (Joao Gante, 2023). Thus, low temperatures are more likely to retain most of the latency benefits from generation via draft models. Additionally, we also observe that temperatures surrounding the peak values always lead to sub-optimal speedups. This is intuitive as the temperature can be seen as an approximate distribution measure. Apart from that, we find that higher temperatures in the surrounding ones usually lead to better results. For example, KD temperature at 0.7 is better than 0.5 when decoding at temperature 0.6 even with the same temperature difference. This highlights another important factor, the diversity of data in KD, for the decoding process.

Using consistent temperatures for KD and decoding leads to better results. Our study reveals that configurations along the diagonals of Figure 2 typically yield the most accelerated decoding speeds. Grids outside the diagonals show pretty large differences with values on diagonals, with a peak relative difference of 24%. This verifies the effectiveness of KD at a consistent temperature. The speedup can be attributed to the alignment of probability distributions when the KD and decoding temperatures are nearly identical or perfectly match. We posit that this alignment facilitates a

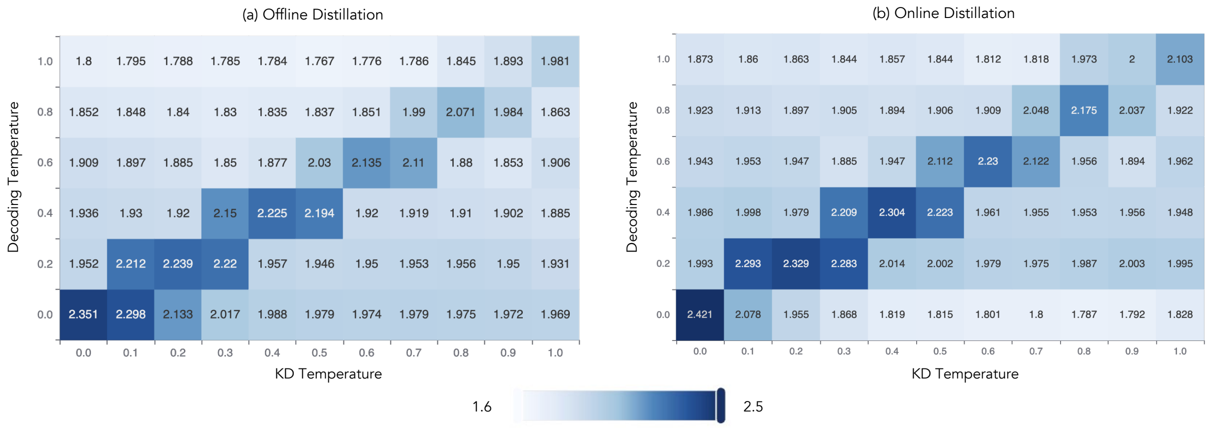

Figure 2: Speedup for different decoding temperatures (y-axis) corresponding to different temperatures during KD (x-axis) for both (a) offline distillation and (b) online distillation for the testing of in-domain Alpaca set.

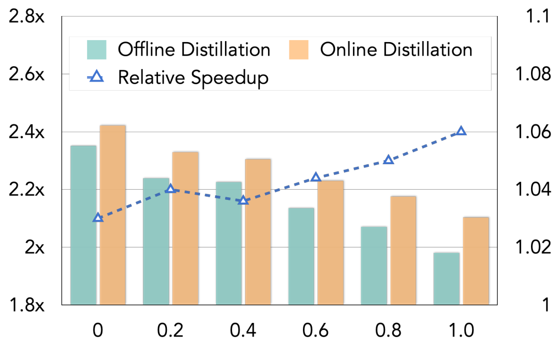

Figure 3: Peak speedup brought by offline distillation and online distillation. The relative speedup for online distillation against offline distillation is depicted in dashed lines.

more efficient decoding process. Interestingly, as the decoding temperature increases, the speedup improvement resulting from this alignment diminishes. Specifically, for offline distillation, the relative improvement transitions from 31% down to approximately 7%. Despite the challenges associated with accelerating speculative decoding at elevated temperatures, employing a uniform KD temperature for decoding — particularly at 1.0 — proves to be more effective than using 0. That being said, the upper right corner of Figure 2 is darker than the upper left corner. This finding further underscores the potential of KD as a remedy for alleviating the difficulty in decoding under high-temperature conditions.

Online distillation is a better KD strategy for speculative decoding compared with offline distillation. Figure 2 illustrates that online distillation consistently outperforms offline distillation across a range of decoding temperatures. This is particularly evident at higher KD temperatures, where the student model benefits from softened probability distributions, allowing for a more nuanced understanding of the teacher's distributions. For better observation, we also plot the peak speedup for every decoding temperature in Figure 3, where the relative speedup of online distillation against offline distillation is in an increasing trend with higher temperatures. Additionally, we find that although online distillation surpasses offline distillation across multiple temperatures, the performance for online distillation at decoding temperature 0 does not align with our expectations, especially with higher KD temperatures. Despite the alignment difference for binary temperature extremes between 1.0 and 0, the richer signal offered by online distillation could be another important factor since decoding at temperature 0 usually entails hard labels.

4.2 Evaluation on Out-of-domain Test Sets

To test whether our observations could be extended to out-of-domain datasets from training sets, we conduct experiments on GSM8K, a dataset focusing on multi-step graduate-school-level mathematical reasoning problems. It differs from the Alpaca training set that focuses more on general domains for everyday tasks. Results are shown in Figure 4.

Generally, the speedup brought by speculative decoding for GSM8K is much larger than that for the Alpaca set. This could seem counterintuitive for an out-of-domain testing set. One potential reason could be that the output for GSM8K consists of *easier* tokens for the draft model to predict. Therefore, the acceptance rate is much higher for target models, which leads to a larger speedup. We found that the number of tokens gen-

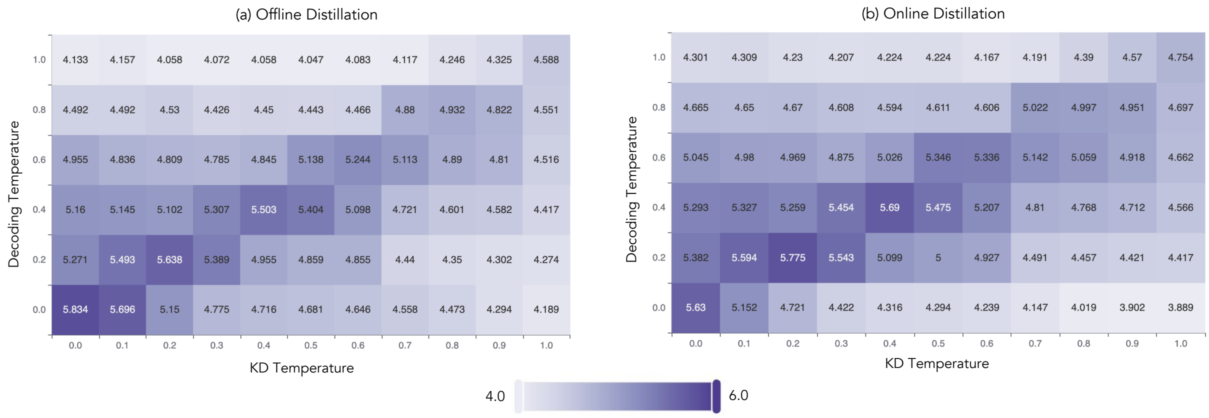

Figure 4: Speedup for different decoding temperatures (y-axis) corresponding to different temperatures during KD (x-axis) for both (a) offline distillation and (b) online distillation for the testing of out-of-domain GSM8K set.

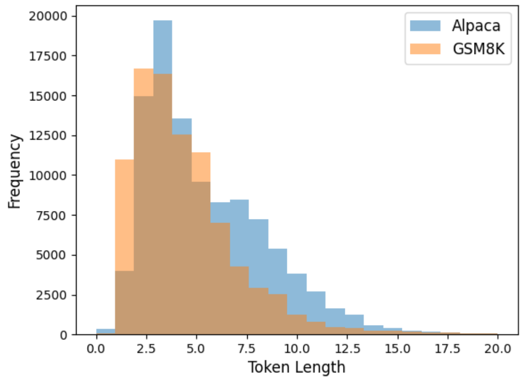

Figure 5: The distribution of token length and the frequencies for both Alpaca and GSM8K test sets.

erated for the Alpaca set $(18,716)$ is much larger than that of GSM8K $(11,130)$, around 68% more than GSM8K, indicating the diversity in decoding processes. We also plot the distribution of token length for generation outputs in Figure 5. Intuitively, length can be seen as an approximate of the difficulty for that token. We observe that token length distribution for Alpaca is leaning towards longer tokens. This phenomenon sheds light on differentiating tokens of difficulties and designing corresponding strategies (Shen et al., 2024) or employing Mixture-of-Experts structures (Shazeer et al., 2017) at a token level.

The overall trend for GSM8K set at different decoding temperatures with KD settings is similar to Alpaca sets. Apart from this, we observe two other notably different phenomena. First of all, **the absolute differences in speedup across various temperatures for GSM8K are significantly larger than that for Alpaca.** For example, with a KD temperature of 0, the relative speedup difference achieved on the Alpaca set is around 30% when the decoding temperature is set to 0 and 1.0, respectively. However, this value increases to 42% for the GSM8K set. This pronounced variance indicates *a stronger sensitivity* to the decoding temperature in the GSM8K set. Such sensitivity may be attributed to the nature of the mathematical reasoning tasks, which perhaps rely more critically on certain temperature thresholds to achieve optimal speculative decoding performance. Additionally, we find that **decoding at temperature 0 with online distillation is particularly slow.** For one thing, the most aligned and fast choice of training under KD temperature 0 does not yield the best speedup. Also, both offline distillation and online distillation do not yield strong performance at decoding temperature 0. In contrast, offline distillation on the Alpaca set shows positive results.

4.3 Evaluation on Out-of-range Decoding Temperatures

In the previous experiments, we mainly focus on a traditionally recommended temperature range $[0, 1]$ that makes LLMs respond in a human-acceptable way. To further understand the robustness and adaptability of our models, we have conducted additional experiments by evaluating them using out-of-range decoding temperatures. Specifically, we have expanded our evaluation to include decoding temperatures of 1.5 and 2.0, which are beyond the commonly used upper limit.

As illustrated in Table 2, we observe several notable phenomena in the performance of both the Alpaca and GSM8K test sets when the decoding temperature is set to these higher values of 1.5 and 2.0. **First of all, we find a similar decreasing trend of speedup when the decoding temperature gets higher.** Specifically, we witness a relative difference of around 15% of decoding at temperature 2.0 compared with 1.5. We also obtain the same observation where the speedup brought for offline distillation is larger than that for online distillation. However, the effect brought by different KD paradigms does not offset decoding temperatures. The effect of decoding temperatures tends to have different representations concerning datasets. Notably, GSM8K seems to have larger speedup differences for temperatures 1.5 and 2.0. This is because GSM8K has a higher speedup as baselines.

Interestingly, the data reveals a distinctive *U-curve* in the relationship between KD temperature and decoding speedup. For instance, with the Alpaca test set at a decoding temperature of 1.5, the speedup incrementally declines from 1.52x at KD temperature 0 to 1.45x at KD temperature 0.4, before ascending back to 1.58x at KD temperature 1.0. For one thing, increasing data diversity during KD training still helps for out-of-range and higher decoding temperatures, which might be caused by the somewhat approaching distributions with target models. However, speedup with KD temperature 0 suggests that generation with fixed configurations holds a special meaning, potentially due to the alignment of distributions between the student and teacher models at this initial point.

4.4 Temperature-aware Recipe for Speculative Decoding

In our prior investigations (as detailed in § 4.1), we establish that decoding at higher temperatures presents challenges. However, we also discover that KD can act as a promising remedy when training models under consistent temperature conditions. In this section, we propose a temperature-aware recipe for speculative decoding inspired by Chang et al. (2023). Our approach employs a simple and intuitive *data-centric composition* strategy, which represents an initial step toward enhancing decoding speed.

Specifically, we first manually identify the top-k best-performing KD temperatures for the target decoding temperature from Figure 2 motivated by the following: (i) Values that approximate the best-performing temperature tend to align more with the target model's distribution; (ii) Diversity in training data for KD further boosts the performance. The selected temperature values are then used for KD in both settings for generation with teacher model and online student model generation. The detailed temperature configurations and experiment results are shown in Table 3.

The composition data for KD are all chosen from the generation of the surrounding peak temperatures. On both Alpaca and GSM8K sets, we observe huge improvements in speedup, achieving an increase of 12%-20%. Interestingly, a decoding temperature of 0.8 with composition yields higher speedups than the higher temperature of 1.0, suggesting that the influence brought by compositional data generation can fully make up for the slow speed when decoding at high temperatures. For the GSM8K dataset, similar trends are observed with even greater speedup values. For instance, with offline distillation and a KD temperature set of $\{0.9, 0.8, 0.7\}$, we achieve the highest reported speedup of 5.62 with an impressive acceptance rate of 89.5%. Additionally, the observed differences in speedup gains between offline distillation and online distillation methods indicate that the former may be more amenable to training data composition strategies. These strategies, which leverage a set of temperatures rather than a single temperature, introduce a more nuanced control over the generated data's variability and quality. This granularity appears to be particularly beneficial for offline distillation, potentially due to the method's intrinsic reliance on the data itself as the primary source of knowledge transfer, which is well aligned with the black-box offline distillation.

5 Related Work

Speculative Decoding The sequential decoding strategy that is prevalently used in autoregressive Transformers (Vaswani et al., 2017) brings latency in real-world servings. To reduce the latency and accelerate decoding speed, the idea of parallel decoding was initially explored in various works (Stern et al., 2018; Ghazvininejad et al., 2019), with strict constraints and deviated distributions. Speculative decoding (Leviathan et al., 2023; Chen et al., 2023a) brings success in reducing the inference latency of LLMs, some recent works (Xia et al., 2024) have attempted to further improve speculative decoding by reducing the rejection rate of candidate tokens. Specifically, Predictive Pipeline Decoding (Yang et al., 2023) was proposed at first to incorporate early exit (Schuster et al., 2022) into the decoding process. Another line of work is to leverage the target model for the self-drafting process, such as Draft&Verify (Zhang et al., 2023), Medusa (Cai et al., 2024), and Speed (Hooper et al., 2023). Tree attention is also explored, where multi-

KD Temp.	Offline Distillation						Online Distillation					
	0	*0.2*	*0.4*	*0.6*	*0.8*	*1.0*	*0*	*0.2*	*0.4*	*0.6*	*0.8*	*1.0*
Alpaca test set												
w/ decoding temp. 1.5	1.58x	1.55x	1.53x	1.52x	1.56x	1.60x	1.52x	1.49x	1.45x	1.50x	1.53x	1.58x
w/ decoding temp. 2.0	1.27x	1.25x	1.23x	1.26x	1.30x	1.35 x	1.22x	1.19x	1.16x	1.21x	1.23x	1.27x
GSM8K test set												
w/ decoding temp. 1.5	3.50x	3.48x	3.44x	3.47x	3.52x	3.59x	3.41x	3.36x	3.30x	3.34x	3.42x	3.48x
w/ decoding temp. 2.0	3.11x	3.09x	3.07x	3.04x	3.05x	3.07x	3.02x	2.93x	2.90x	2.92x	2.96x	3.03x

Table 2: Performance with out-of-range decoding temperatures on two KD settings with both Alpaca and GSM8K test set.

Methods	Decoding	KD temp.		Alpaca		GSM8K	
	temp.	Alpaca	GSM8K	*Acc. Rate*	*Speedup*	*Acc. Rate*	*Speedup*
Offline distillation	1.0	1.0	1.0	80.6	1.98x	86.1	4.59x
	0.8	0.8	0.8	81.9	2.07x	87.3	4.93x
w/ composition	1.0	{1.0, 0.9, 0.8}	{1.0, 0.9, 0.8}	83.0	2.23x	88.7	5.28x
	0.8	{0.9, 0.8, 0.7}	{0.9, 0.8, 0.7}	83.6	**2.34x**	89.5	**5.62x**
Online distillation	1.0	1.0	1.0	82.2	2.10x	87.1	4.75x
	0.8	0.8	0.8	82.6	2.18x	87.9	5.00x
w/ composition	1.0	{1.0, 0.9, 0.8}	{1.0, 0.9, 0.8}	83.5	2.27x	88.5	5.20x
	0.8	{0.9, 0.8, 0.7}	{0.9, 0.8, 0.7}	83.7	**2.33x**	88.9	**5.41x**

Table 3: Performance with data composition on two KD settings. Acceptance rate and speedup are reported for both in-domain and out-of-domain datasets.

ple candidates during drafting are taken into consideration (Miao et al., 2023). Cascaded drafting process (Spector and Re, 2023; Chen et al., 2023b) is also invented to reduce drafting latency. However, almost all of the previous works only investigate the coarse-grained effect brought by generation configurations, such as temperature. For example, CSDecoding (Chen et al., 2023b) and SpecInfer (Miao et al., 2023) only explore greedy decoding for testing. Our work mostly relates to the work that leverages knowledge distillation (Zhou et al., 2023; Liu et al., 2023), with a focus on temperature-centric investigation for instruction-tuned KD draft models.

Knowledge Distillation for LLMs Knowledge distillation (KD) (Hinton et al., 2015) is a widely used model compression technique, aiming at training a student model with the guidance of a teacher model (Gou et al., 2021). The student model emulates the teacher models' behavior on downstream tasks. Standard KD methods are approximately minimizing the generation distribution of the student and the teacher. This is achieved by using the teacher's output at each time step as supervision (Sanh et al., 2019) or direct training on the teacher's generated texts (Kim and Rush, 2016). With the emergence of LLMs, more techniques were invented for KD of LLMs, such as using reversed KL Divergence (Gu et al., 2024) or other variants of KLD (Agarwal et al., 2023; Wen et al., 2023). In this work, since we are targeting temperature-centric investigation of KD for speculative decoding, we only explore the two standard KD settings, i.e., black-box SeqKD (Kim and Rush, 2016), and online data generation that targets better KD for LLMs (Agarwal et al., 2023).

6 Conclusion

In this paper, we have presented a comprehensive investigation into the impact of *temperature* on speculative decoding, particularly within the context of knowledge distillation (KD), for large language models (LLMs). Through a series of meticulous experiments utilizing the Llama series as both target and draft models, we have explored the nuanced interplay between temperature settings during KD and their consequent effect on speculative decoding's efficiency and efficacy. Apart from offering empirical findings, we also propose a practical strategy to enhance speculative decoding's performance by leveraging temperature-centric training data assembly. By presenting this work, we aspire to facilitate future works on diverse generation configurations for speculative decoding, and exploring theoretical understanding of the multifaceted relations in between.

Limitations

We discuss the limitations of this work in the following aspects:

1. **Scope of the paper:** The factor of temperature for speculative decoding is an important aspect to investigate. While we investigated a general setting of knowledge distillation, we were not able to explore other settings due to limited computation resources.

2. **Empirical analysis:** This study is an empirical investigation of the effect brought by different temperatures in speculative decoding. We interpret the conclusions and findings largely based on observations at hand, without solid theoretical foundations. Future works are encouraged to explore this direction.

3. **Preliminary approach:** This study attempts to understand and accelerate speculative decoding at higher temperatures. We propose an empirical solution for data composition that has proven effective in our tests. However, our primary focus was not on developing comprehensive algorithms for speedup at higher temperatures. Further work could create more refined and mature solutions in this area.

References

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cogn. Sci.*, 9:147–169.

Rishabh Agarwal, Nino Vieillard, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2023. Gkd: Generalized knowledge distillation for auto-regressive sequence models. *arXiv preprint arXiv:2306.13649*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862.

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

Chung-Ching Chang, David Reitter, Renat Aksitov, and Yun-Hsuan Sung. 2023. Kl-divergence guided temperature sampling. *arXiv preprint arXiv:2306.01286*.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023a. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv: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. *arXiv preprint arXiv:2312.11462*.

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

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Hasan Genc, Kurt Keutzer, Amir Gholami, and Sophia Shao. 2023. Speed: Speculative pipelined execution for efficient decoding. *arXiv preprint arXiv:2310.12072*.

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

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

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

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Ion Stoica, Zhijie Deng, Alvin Cheung, and Hao Zhang. 2023. Online speculative decoding. *arXiv preprint arXiv: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. *arXiv preprint arXiv:2305.09781*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2022. Efficiently scaling transformer inference. *CoRR*, abs/2211.05102.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Shannon Zejiang Shen, Hunter Lang, Bailin Wang, Yoon Kim, and David Sontag. 2024. Learning to decode collaboratively with multiple language models. *arXiv preprint arXiv:2403.03870*.

Benjamin Spector and Chris Re. 2023. Accelerating llm inference with staged speculative decoding. *arXiv preprint arXiv:2308.04623*.

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.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv: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, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 10817–10834, Toronto, Canada. Association for Computational Linguistics.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *arXiv preprint arXiv:2401.07851*.

Seongjun Yang, Gibbeum Lee, Jaewoong Cho, Dimitris Papailiopoulos, and Kangwook Lee. 2023. Predictive pipelined decoding: A compute-latency trade-off for exact llm decoding. *arXiv preprint arXiv:2307.05908*.

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

Ming Zhong, Chenxin An, Weizhu Chen, Jiawei Han, and Pengcheng He. 2024. Seeking neural nuggets: Knowledge transfer in large language models from a parametric perspective. In *The Twelfth International Conference on Learning Representations*.

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. *arXiv preprint arXiv:2310.08461*.

A KD Algorithm

In this section, we give the detailed algorithm of the KD training setting used in this paper.

Algorithm 1: Online Distillation Algorithm

Input: Target model $\mathcal{M}_t$, Draft model $\mathcal{M}_d^\theta$, Data set containing (input, output) pairs
Output: Distilled draft model $\mathcal{M}_d^\theta$
Hyperparameters: Data fraction from online generation $\lambda \in [0,1]$, Temperature $\tau \in [0,1]$, loss ratio $\gamma \in [0,1]$

```
for k in 0, ..., n do
    Generate a random value μ ∈ (0, 1)
    if μ ≤ λ then
        sample inputs x from X and generate
          outputs y by M_d^θ(·|x) with temperature τ
    end
    else
        sample inputs x from X and outputs y from
          Y
    end
    Update θ to minimize
      L = L_lm + γD(M_t||M_d^θ(y|x))
end
```

B Implementation Details

Data Formulation for Alpaca Dataset For knowledge distillation, we instruction-tuned the model on the Alpaca dataset. Specifically, for each data sample in the dataset with triple "instruction-input-output", we use the following template to curate input for training:

If the elements in the triple are complete, we use the following template:

`Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. ###Instruction:{instruction} ### Input:{input}### Response:`

If there is only "instruction" for the data sample without "input", the above template will be simplified as:

`Below is an instruction that describes a task. Write a response that appropriately completes the request.### Instruction:{instruction}### Response:`

Implementation Details for KD For online distillation, we set the batch size to 8, learning rate to 3e-5, maximum length of input to 512. The training process continues for 30 epochs with 200,000 steps in total. It takes around 30 hours to finish. For offline distillation, it takes 8 hours to finish.

Implementation Details for Evaluation We set the maximum decoding length to 128 due to the limit in A100 40G' GPU memory. Each evaluation corresponding to KD temperatures and decoding temperatures requires around $12h$ to run on the A100 GPU with batch size 1.

C Detailed analysis for Section 4.1

Speedup is hard to get offset with longer KD steps. According to our observation, the optimal performance is achieved when the decoding temperature and KD temperature align with each other. To further understand the improvement in speedup regarding the temperatures, we study the relation with KD steps in Figure 6. We consider a rather extreme setting where the decoding temperature is set as 1.0 with KD temperatures 0 and 1.0. During the initial stages of knowledge distillation, the two curves representing different temperature settings exhibit rapid growth and are relatively close to each

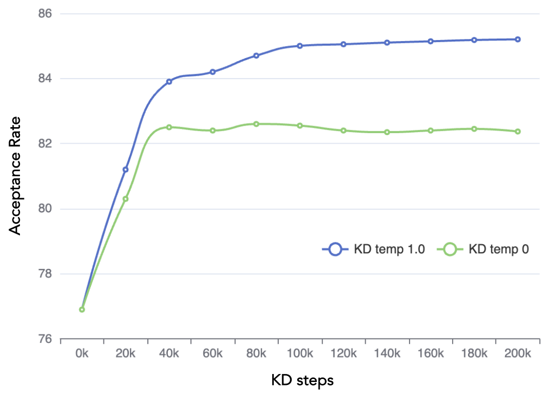

Figure 6: Acceptance rate of different KD temperatures for decoding at temperature 1.0 regarding KD steps on the Alpaca test set.

other. As the KD process progresses, the curve with KD temperature 1.0 diverges significantly from the other and the acceptance rate still steadily increases. As the KD process gradually approaches the end, the curve with KD temperature 1.0 achieves higher speedup and continues to show an upward trend, whereas the other temperature curve plateaus with a lower acceptance rate.

Phenomenon of symmetric temperature configurations. Intuitively, we might expect that distilling from a teacher with temperature τ_1 and then using decoding temperature τ_2 can behave similarly to distilling with temperature τ_2 and decoding with temperature τ_1. This phenomenon could be referred to as diagonals (from upper left corner to lower right) in Figures 2. We find that symmetric temperature settings do bring similar speedups. However, decoding at lower temperatures is still faster than at higher temperatures.