AdaEDL: Early Draft Stopping for Speculative Decoding of Large Language Models via an Entropy-based Lower Bound on Token Acceptance Probability

Sudhanshu Agrawal **Wonseok Jeon** **Mingu Lee**
Qualcomm AI Research[1]
{sudhagra, wjeon, mingul}@qti.qualcomm.com

Abstract

Speculative decoding [1] is a powerful technique that attempts to circumvent the autoregressive constraint of modern Large Language Models (LLMs). The aim of speculative decoding techniques is to improve the average inference time of a large, *target* model without sacrificing its accuracy, by using a more efficient *draft* model to propose draft tokens which are then verified in parallel. The number of draft tokens produced in each drafting round is referred to as the draft length and is often a static hyperparameter chosen based on the acceptance rate statistics of the draft tokens. However, setting a static draft length can negatively impact performance, especially in scenarios where drafting is expensive and there is a high variance in the number of tokens accepted. **Ada**ptive **E**ntropy-based **D**raft **L**ength (**AdaEDL**) is a simple, training and parameter-free criteria which allows for early stopping of the token drafting process by approximating a lower bound on the expected acceptance probability of the drafted token based on the currently observed entropy of the drafted logits. We show that AdaEDL consistently outperforms static draft-length speculative decoding by 10%-57% as well as other training-free draft-stopping techniques by upto 10% in a variety of settings and datasets. At the same time, we show that AdaEDL is more robust than these techniques and preserves performance in high-sampling-temperature scenarios. Since it is training-free, in contrast to techniques that rely on the training of dataset-specific draft-stopping predictors, AdaEDL can seamlessly be integrated into a variety of pre-existing LLM systems.

1 Introduction

Large Language Models (LLMs) have been shown to have impressive performance on a variety of tasks including creative writing, summarization, and translation [2]. In particular, in recent years, several *foundation models* such as Llama2 [3], Llama3 [4], GPT-4 [5], and Claude [6] have shown to exceed expectations and generalize to coding [7], display agentic abilities [8], interpret images [9], and more. In all such systems, an LLM's job remains the same - prediction of the next token via autoregressive generation. Autoregressive generation is fundamentally sequential in nature, since the prediction of the next token can only occur once the previous token has been generated. This reduces the ability of an LLM to parallelize, creating a bottleneck in the maximum number of generated tokens per second (TPS).

Speculative decoding techniques attempt to introduce parallelism to this system. Consider a scenario where the objective is to perform inference on a *target model*, say Llama2-7B. A smaller *draft model*

[1] Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

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

with the same tokenizer as the target model, say TinyLLama-1B [10] is then chosen. Given a prompt, the draft model is allowed to run autoregressively, producing a set of candidate draft tokens. These tokens are then consumed by the target model which produces logits for each draft token, representing a probability distribution. Rejection sampling [1] then guarantees that the draft tokens which are *accepted* via this process will preserve the distribution of the original target model. Thus, by running a small model autoregressively and a large model in parallel, the system as a whole experiences an increase in average token rate.

However, a limiting factor of this system is that the *number* of draft tokens produced, referred to as the draft length (DL), if fixed over multiple rounds of drafting, can reduce the average token rate. This may be caused due to over-utilization of a poorly performing draft model, or symmetrically, because the draft model is under-utilized which does not allow the speculative decoding system to reach its maximum possible performance. For example, since most target models are large foundation models, designed to have high accuracy on a variety of tasks, it is likely that a smaller draft model finetuned to match the target model distribution for a particular task may have varying levels of accuracy when the target model switches tasks. In such scenarios, one can see a high variance in the number of accepted draft tokens. That is, in some drafting rounds, almost all the draft tokens are accepted, whereas in some, almost all are rejected. Figure 1 considers the creative writing task from the Dolly-15k dataset [11]. For this dataset, we see that for a standard speculative decoding system operating with various static draft lengths, the number of accepted tokens follows a normal distribution, with num-accepted-tokens taking on almost every value from 0 to max-draft-length with varying frequency. We include similar figures for the CNN-DM (summarization) [12] and WMT-19 (German-English translation) [13] datasets along with the details to set up these speculative decoding systems in Appendix A Figures 7a, 7b.

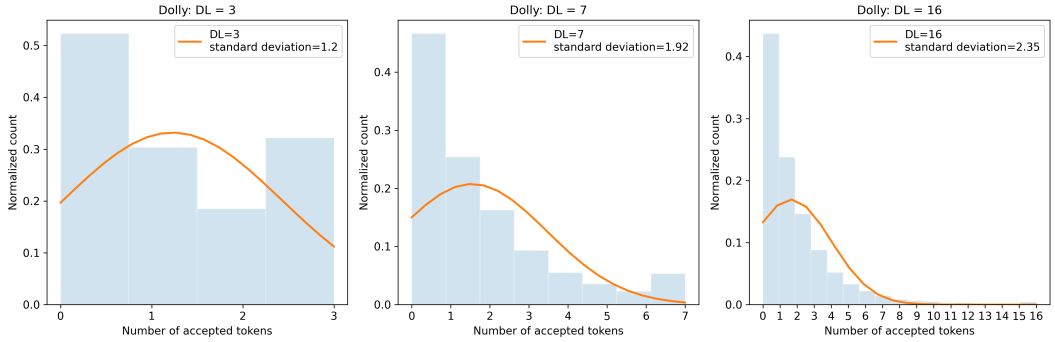


Figure 1: The number of accepted tokens across drafting rounds displays a high variance, leading to under or over-utilization of the draft model in static draft length speculative decoding methods.

This motivates the need for an **adaptive draft length** speculative decoding system where the draft length at every drafting round can be determined on-the-fly. AdaEDL approaches this problem with a *go-no-go strategy*: while drafting tokens, at every iteration, AdaEDL establishes a draft-stopping criteria by approximating a lower bound on the expected acceptance probability of the drafted token by using the entropy of the draft model logits at that iteration. If the criteria is satisfied, drafting stops and verification by the target model is performed. We provide a theoretical basis to our proposed draft-stopping criteria formulation, deriving how it relates to the acceptance probability of the draft model. To validate our approach, we perform experiments across various maximum draft length settings, across multiple datasets and sampling temperature settings, and for various target and draft model choices, showing that this new draft-stopping strategy is more effective and robust than draft-stopping strategies which simply use the probability value of the most-likely token as a draft-stopping criteria - for example, those used in BiLD [14] and Draft & Verify [15]. Indeed, AdaEDL may be used to complement either of these algorithms. At the same time, our proposed system avoids the need to train an independent network to act as a binary classifier for early draft-stopping for a specific model and task, such as the approaches followed by SpecDec++ [16] and DISCO [17], which makes AdaEDL a simple and straightforward improvement to boost the token rate of speculative decoding LLM systems.

2 Problem setting and existing methods

Following a notation similar to the original speculative decoding formulation, let TM be the target model whose inference we are trying to accelerate. Let DM be a more efficient approximation of this target model, referred to as the draft model. Let us denote the t^{th} token in the prompt by x_t. Then the probability distribution of TM at any given token x_t may be denoted as $p_{TM}(x_t|x_{<t})$. Similarly, the probability distribution of the draft model, DM may be denoted as $p_{DM}(x_t|x_{<t})$. As noted in [1], several strategies such as nucleus sampling, top-k sampling, and others may each be viewed as sampling from an adjusted probability distribution. For notational simplicity, we refer to these probabilities, adjusted or not, as $p_{TM}(x)$ and $p_{DM}(x)$. In this notation, we can now describe the various decoding techniques:

Autoregressive decoding In this baseline technique, we only consider the target model and at every iteration, a new token is sampled as

$$x_t \sim p_{TM}(\cdot|x_{<t})$$

Speculative decoding The system first allows a draft model to consume the tokens $x_{<t}$. The draft model then autoregressively generates L draft tokens $d_1, d_2, \cdots d_L$, where L is the maximum draft length. This can be viewed as sampling a token

$$d_i \sim p_{DM}(\cdot|x_{<t}, d_1, \cdots, d_{i-1})$$

After drafting, the target model evaluates these draft tokens in parallel, producing probabilities $p_{TM}(x)$. The draft token d_i is *accepted* if $p_{DM}(x) \leq p_{TM}(x)$, producing a token $x_i = d_i$. Otherwise, it is *rejected* with probability $1 - p_{TM}(x)/p_{DM}(x)$ and a new token x_i is sampled from an adjusted distribution $x_i \sim p'_{TM}(x) = norm(max(0, p_{TM}(x) - p_{DM}(x)))$. A token x_i sampled via this method, referred to as rejection sampling, is guaranteed to follow the probability distribution of the original target model, that is, $x_i \sim p_{TM}(x)$ [1].

To further improve this process, the draft length L may be made *adaptive*, to avoid the cost of drafting tokens that are likely to be rejected anyway.

Adaptive draft length via maximum confidence speculative decoding Some systems consider the top-1 probability of the drafted logits. That is, at a given drafting stage, the system considers the token with the highest probability $\max_x p_{DM}(x|x_{<t}, d_1, \cdots, d_{i-1})$. If this value, the *maximum confidence* among any possible token, is less than a threshold, λ, the system is considered to be *under-confident* and drafting stops.

$$\max_x p_{DM}(x) < \lambda$$

This is a simple method to avoid wastage during the drafting phase and is effectively employed alongside other techniques in Draft & Verify [15] and BiLD [14]. However, this scheme fails to take into account the overall probability distribution during drafting. This motivates the need for a new drafting stopping mechanism that considers the probabilities of all possible tokens while making a go-no-go decision.

Adaptive draft length via a trained predictor Several works involve the training of a small network to act as a predictor to determine optimal draft lengths, for example, SpecDec++ [16] which trains a ResNet and DISCO [17] which trains a FFN as a predictor for early draft stopping. AdaEDL is distinct from these method and specifically aims to be training and parameter-free, similar to draft stopping via maximum confidence, to allow for greater generalization across datasets and models.

3 Adaptive entropy-based draft length speculative decoding (AdaEDL)

AdaEDL operates in the same setup as the above adaptive draft length techniques but instead, establishes a stopping criteria using an entropy-based lower bound on the token acceptance probability. If we consider the probability distribution of the draft model, $p_{DM}(x)$ and its corresponding entropy $H_{DM}(x)$, we see that $1 - \sqrt{\gamma H_{DM}(x)}$ serves as an approximate lower bound on the expected acceptance rate. This allows us to formulate a drafting system where drafting is stopped if this criteria falls below a threshold λ, indicating that the expected acceptance rate is also below this threshold:

$$1 - \sqrt{\gamma H_{DM}(x)} < \lambda$$

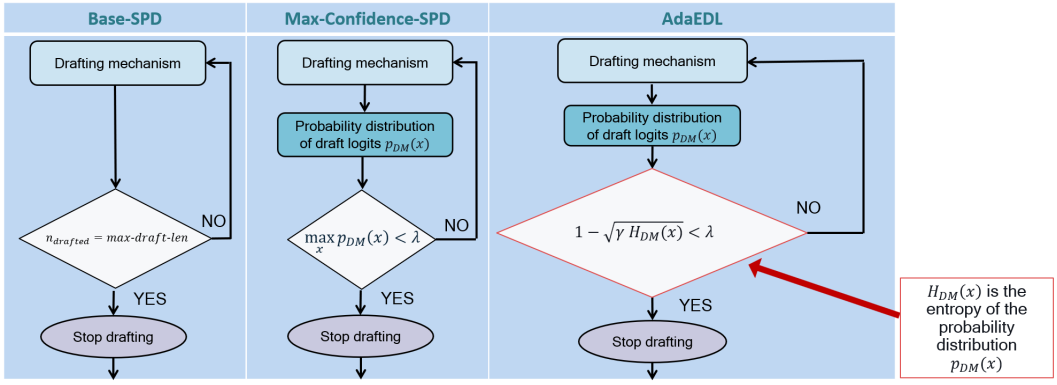


Figure 2: AdaEDL performs adaptive early-draft-stopping via an entropy-based lower bound on the expected acceptance rate.

Algorithm 1 Dynamic updates for stopping threshold λ

$L \leftarrow$ max_draft_length
$n_{drafted} \leftarrow$ num_drafted_tokens
$n_{acc} \leftarrow$ num_accepted_tokens
$i \leftarrow$ cur_drafting_round
$\lambda \leftarrow$ cur_stopping_threshold
while $i \geq 1$ **do**
    $AR_i \leftarrow n_{acc}/n_{drafted}$ ▷ Calculate the current acceptance rate
    $AR \leftarrow \beta_1 AR + (1-\beta_1) AR_i$
    if $AR < \alpha$ **then** ▷ Not yet meeting the target acceptance rate
        $\lambda' \leftarrow \lambda + \epsilon$
    else if $n_{acc} \neq L$ **then** ▷ Not yet drafting max possible tokens
        $\lambda' \leftarrow \lambda - \epsilon$
    else
        $\lambda' \leftarrow \lambda$
    end if
    $\lambda \leftarrow \beta_2 \lambda + (1-\beta_2)\lambda'$ ▷ Update the stopping threshold
end while

Hyperparameters: α = target acceptance rate, ϵ = step size, β_1, β_2 are used to calculate the exponential moving averages. Refer to Section 4.4 for additional details.

Here, $\gamma > 0$ is a hyperparameter. We include a detailed derivation of this lower-bound approximation in Appendix B. Additionally, we visualize AdaEDL alongside the baseline speculative decoding systems mentioned above in Figure 2, highlighting the novel early draft-stopping criteria we introduce.

Improving λ via dynamic updates The stopping threshold λ can be further optimized by making it responsive to the current acceptance rate statistics of the system. In particular, we aim to achieve a target acceptance rate α - increasing the stopping threshold if it is not currently being reached and decreasing it otherwise. By maintaining an exponential moving average of the acceptance rate and using it to update the stopping threshold in this manner, we can achieve better performance over longer generations. We modify the threshold update strategy described in [15] and describe it in Algorithm 1.

We further discuss the hyperparameters introduced here, γ, λ, β_1, β_2, α, and ϵ, their typical values and ranges, as well as the sensitivity of AdaEDL to these hyperparameters in Section 4.4.

4 Experimental Results

In all of the following results, **AdaEDL** refers to our proposed method, described in in Section 3 with dynamically updated stopping thresholds as described by Algorithm 1. **Max-Confidence-SPD**

refers to a speculative decoding system which implements the early draft-stopping strategy based on the token with the highest probability described in Section 2, which is further improved by using the same dynamic threshold strategy to have a fair comparison with AdaEDL. **Base-SPD** refers to vanilla speculative decoding with no early draft-stopping strategy employed and **Autoregressive** refers to standard autoregressive decoding of LLMs which is the baseline that speculative decoding systems attempt to improve on.

We evaluate the performance of AdaEDL across various datasets and tasks: Dolly-15k (creative writing) [11], WMT-19 (German-English translation) [13], and CNN-DM (summarization) [12]. The entire Dolly test dataset (708 samples), the entire WMT-19 test dataset (600 samples), and the first 1000 samples of the CNN-DM test dataset were used to ensure a large sample size. All token rate (TPS) numbers reported are calculated as the averages over all samples in each dataset. All experiments were performed on a single NVIDIA A100 with 80GB of GPU memory in FP32 precision. We acknowledge here that results may vary on different hardware which may lead to faster or slower inference of a draft or target model and in particular, their relative speed. We discuss this point in Appendix B.2.

We show in the following experiments that AdaEDL will always either exceed or match the performance of other decoding systems while also proving less sensitive to chosen hyperparameters and robust to the system's settings.

4.1 Performance with fast finetuned draft models

In the following experiments, we use Llama2-7B as the target model with sampling temperature set to 0.7. The draft model, Llama2-Drafter-115M, is a 115M parameter model, distilled and finetuned using direct alignment [18] to closely match the target model distribution . As a result, even Base-SPD results in higher acceptance rates than one would observe from an off-the-shelf draft model.

In Table 1, we indicate the performance of AdaEDL across the various datasets and tasks mentioned above - Dolly-15k, WMT-19, and CNN-DM and report the average token rate on each of these datasets. We observe that AdaEDL consistently matches or beats both Max-Confidence-SPD and Base-SPD on all tasks by a competitive margin. In particular, AdaEDL has a significant advantage in the open-ended CNN-DM summarization task.

Table 1: Performance of AdaEDL vs Max-Confidence-SPD vs Base-SPD for various maximum draft lengths and datasets. Target model = Llama2-7B, draft model = Llama2-Drafter-115M, sampling temperature = 0.7.

Max DL = 16	CNN-DM	Dolly-15k	WMT-19
Autoregressive	25.74	29.02	29.80
Base-SPD	36.30	32.10	22.30
Max-Confidence-SPD	49.50	55.80	43.70
AdaEDL	**54.10**	**56.10**	**43.90**
Max DL = 7	CNN-DM	Dolly-15	WMT-19
Autoregressive	25.74	29.02	29.80
Base-SPD	51.50	47.60	32.70
Max-Confidence-SPD	53.50	56.60	45.10
AdaEDL	**56.90**	**57.10**	**45.20**
Max DL = 3	CNN-DM	Dolly-15k	WMT-19
Autoregressive	25.74	29.02	29.80
Base-SPD	54.10	54.70	40.90
Max-Confidence-SPD	53.10	55.70	42.50
AdaEDL	**55.70**	**55.80**	**45.00**

4.2 Performance of AdaEDL with expensive draft models

A smaller draft model may be faster at drafting, but sacrifices acceptance rate. Symmetrically, a larger draft model may lead to higher acceptance rates, but lower overall inference speed due to its own cost of inference.

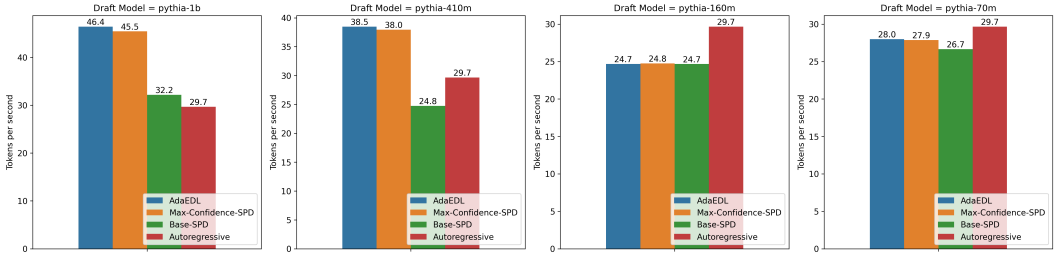


Figure 3: Target model = Pythia-6.9B with various $|TM|/|DM|$ ratios demonstrates that AdaEDL opens up the possibility of using speculative decoding with much larger draft models. Max draft length = 7, sampling temperature = 1.0, dataset = CNN-DM.

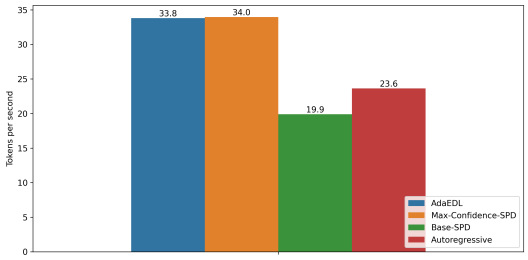


Figure 4: Target model = LLama2-7B, draft model = TinyLLama-1B, max draft length = 7, sampling temperature = 1.0, when used in an adaptive draft length decoding scheme, surpass autoregressive and Base-SPD performance. Dataset = CNN-DM.

4.2.1 Pythia

The Pythia family of models [19] consists of models of sizes ranging from 70M to 12B parameters. As a result, they represent an ideal set of models to test the performance of various decoding systems when the ratio of the target model size to the draft model size is varied - that is, when $\frac{|TM|}{|DM|}$ is varied.

Figure 3 sets Pythia-6.9B as the target model and considers the performance of Pythia-1B, Pythia-410M, Pythia-160M, and Pythia-70M as draft models for the CNN-DM (summarization) task. We demonstrate that adaptive draft length techniques enable us to use upto $10\times$ larger draft models in a scenario where otherwise, the maximum token rate achievable with Base-SPD would have been 32.2 tokens per second (TPS). We see that when Pythia-160M or Pythia-70M is used as the draft model, autoregressive decoding outperforms all speculative decoding methods with a token rate of 29.7 TPS. Without adaptive draft length techniques, the maximum token rate achievable through Base-SPD is 32.2 TPS with Pythia-1B as a draft model for an 8% speedup. However, when Pythia-1B is used to draft for Pythia-6.9B with AdaEDL, the system achieves a token rate of 46.4 TPS for a 56% speedup. Thus, AdaEDL enables speculative decoding in a scenario where autoregressive decoding would normally be the candidate method. This also opens up the possibility of *finetuning* larger draft models, which would presumably lead to higher acceptance rates without sacrificing performance if AdaEDL were to be enabled. We believe that this is a promising direction of investigation for future work.

4.2.2 TinyLlama

Motivated by the results in Section 4.2, Figure 4 shows the performance of various decoding methods when the target model is Llama2-7B and the draft model is a standard 1B model - TinyLlama-1B [10]. We see that in this case, AdaEDL and Max-Confidence-SPD increase the token rate by 43% as compared to autoregressive decoding, while Base-SPD negatively impacts performance, reducing the token rate by 16%.

4.3 Performance of AdaEDL across sampling temperatures

In cases where the target distribution is difficult to predict, such as when the chosen sampling temperature is high, we see that Base-SPD is only able to produce modest gains in token rate -

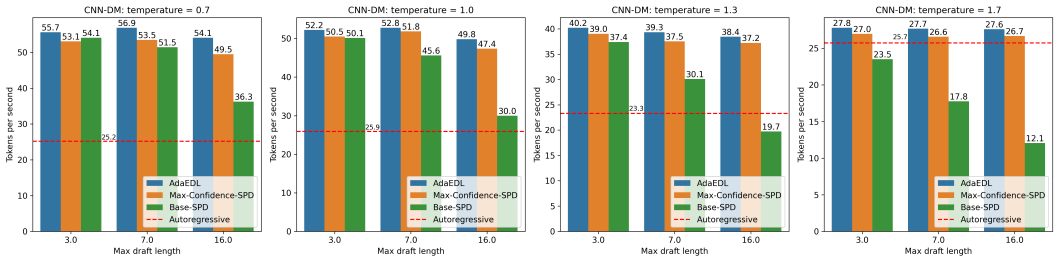


Figure 5: AdaEDL boosts token rate even in high temperature scenarios where Base-SPD is insufficient. Target model = Llama2-7B, draft model = Llama2-Drafter-115M, dataset = CNN-DM.

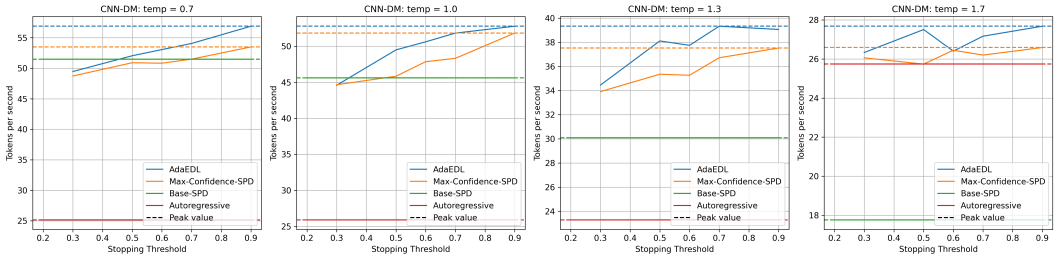


Figure 6: AdaEDL performs consistently better than Max-Confidence-SPD even at suboptimal λ. Target model = Llama2-7B, draft model = Llama2-Drafter-115M, dataset = CNN-DM, max draft length = 7.

sometimes even resulting in poorer performance than autoregressive decoding. In Figure 5 we see that as we increase the sampling temperature from 0.7 to 1.7 on the CNN-DM dataset, the token rate of Base-SPD drops 57% (for draft length 3), and ends up being lower than standard autoregressive decoding. On the other hand, even at high sampling temperatures, AdaEDL provides an 8% boost in token rate over the autoregressive baseline. AdaEDL also consistently outperforms the other 3 decoding methods across lower sampling temperatures and across maximum draft length settings. We perform similar experiments on the Dolly-15k and WMT-19 datasets in Appendix C Figures 8a, 8b, and see similarly consistent performance from AdaEDL.

4.4 Controllability and sensitivity of AdaEDL to hyperparameters

Entropy factor (γ) In all our experiments, we set $\gamma = 0.2$ in the equation $1 - \sqrt{\gamma H_{DM}(x)}$. We see that this simple approximation described in Appendix B, when coupled with the dynamic stopping threshold strategy described in Algorithm 1, produces strong results. Further hyperparameter search may be possible to improve this value for a particular dataset and we observe that values in the range $\gamma \in (0, 1)$ typically work best. It may also be possible to dynamically update γ using the observed target model distribution as the system runs. We defer these investigations to future works.

Threshold update hyperparameters (β_1, β_2, ϵ, α) In all our experiments, we set $\beta_1 = 0.5$, $\beta_2 = 0.9$, $\epsilon = 0.01$, and $\alpha = 0.9$, following [15], for the threshold update step for both AdaEDL and Max-Confidence-SPD.

Stopping threshold (λ) The most significant hyperparameter in adaptive draft length methods is the early draft-stopping threshold, λ. In all our experiments, λ is updated dynamically according to the scheme described in Algorithm 1. Regardless, we find that both AdaEDL and Max-Confidence-SPD are sensitive to the initial choice of λ. We hypothesize that this may be due to the fact that many output generations are short in length, which may not result in enough drafting rounds for the system to converge to an optimal threshold. In Figure 6 we see that AdaEDL, even for sub-optimal λ choices, consistently outperforms Max-Confidence-SPD when its λ is also chosen sub-optimally. At the same time, AdaEDL performs better than this baseline if optimal λ are chosen for both methods as we see marked by the dashed line and in Section 4.1. We conduct experiments across datasets (CNN-DM, Dolly-15k, WMT-19), sampling temperatures (0.7, 1.0, 1.3, 1.7), and maximum draft length settings

(3, 7, 16), and observe similar trends in Appendix D Figures 9a, 9b, 9c, showing that AdaEDL can consistently boost token rate without the need for fine-grained hyperparameter search.

5 Conclusion

In this work, we present AdaEDL, an early stopping criteria for drafting in speculative decoding systems which uses the entropy of the draft model to estimate a lower bound on the current token's acceptance rate. We show the efficacy of this new method across datasets, sampling temperatures, draft lengths, and choice of target and draft models, whether fine-tuned or off-the-shelf. AdaEDL boosts the performance of existing speculative decoding systems while also enabling efficient usage of much larger draft models which, if finetuned in future works, could potentially result in even more impressive gains in token rate. AdaEDL is training and parameter-free, is not dependent on a given dataset, and also offers a relaxed choice in hyperparameters, making it a simple, plug-and-play improvement to a variety of pre-existing speculative decoding LLM systems.

References

[1] Yaniv Leviathan, Matan Kalman, and Yossi Matias. *Fast Inference from Transformers via Speculative Decoding*. 2023. arXiv: 2211.17192 [cs.LG]. URL: https://arxiv.org/abs/2211.17192.

[2] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

[3] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL]. URL: https://arxiv.org/abs/2307.09288.

[4] Abhimanyu Dubey et al. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

[5] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[6] Yuntao Bai et al. *Constitutional AI: Harmlessness from AI Feedback*. 2022. arXiv: 2212.08073 [cs.CL]. URL: https://arxiv.org/abs/2212.08073.

[7] Baptiste Rozière et al. *Code Llama: Open Foundation Models for Code*. 2024. arXiv: 2308.12950 [cs.CL]. URL: https://arxiv.org/abs/2308.12950.

[8] Xiao Liu et al. *AgentBench: Evaluating LLMs as Agents*. 2023. arXiv: 2308.03688 [cs.AI]. URL: https://arxiv.org/abs/2308.03688.

[9] Haotian Liu et al. *Visual Instruction Tuning*. 2023. arXiv: 2304.08485 [cs.CV]. URL: https://arxiv.org/abs/2304.08485.

[10] Peiyuan Zhang et al. *TinyLlama: An Open-Source Small Language Model*. 2024. arXiv: 2401.02385 [cs.CL]. URL: https://arxiv.org/abs/2401.02385.

[11] Mike Conover et al. *Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM*. 2023. URL: https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm (visited on 06/30/2023).

[12] Abigail See, Peter J. Liu, and Christopher D. Manning. "Get To The Point: Summarization with Pointer-Generator Networks". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1073–1083. DOI: 10.18653/v1/P17-1099. URL: https://www.aclweb.org/anthology/P17-1099.

[13] Wikimedia Foundation. *ACL 2019 Fourth Conference on Machine Translation (WMT19), Shared Task: Machine Translation of News*. URL: http://www.statmt.org/wmt19/translation-task.html.

[14] Sehoon Kim et al. *Speculative Decoding with Big Little Decoder*. 2023. arXiv: 2302.07863 [cs.CL]. URL: https://arxiv.org/abs/2302.07863.

[15] Jun Zhang et al. *Draft & Verify: Lossless Large Language Model Acceleration via Self-Speculative Decoding*. 2024. arXiv: 2309.08168 [cs.CL]. URL: https://arxiv.org/abs/2309.08168.

[16] Kaixuan Huang, Xudong Guo, and Mengdi Wang. *SpecDec++: Boosting Speculative Decoding via Adaptive Candidate Lengths*. 2024. arXiv: 2405.19715 [cs.CL]. URL: https://arxiv.org/abs/2405.19715.

[17] Jonathan Mamou et al. *Dynamic Speculation Lookahead Accelerates Speculative Decoding of Large Language Models*. 2024. arXiv: 2405.04304 [cs.CL]. URL: https://arxiv.org/abs/2405.04304.

[18] Raghavv Goel et al. *Direct Alignment of Draft Model for Speculative Decoding with Chat-Fine-Tuned LLMs*. 2024. arXiv: 2403.00858 [cs.LG]. URL: https://arxiv.org/abs/2403.00858.

[19] Stella Biderman et al. *Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling*. 2023. arXiv: 2304.01373 [cs.CL]. URL: https://arxiv.org/abs/2304.01373.

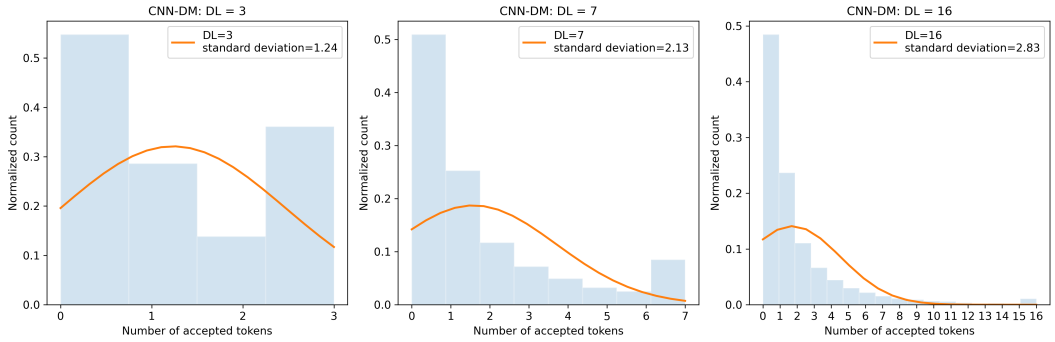
[20] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. 2008. URL: https://api.semanticscholar.org/CorpusID:117035435.

[21] Olivier Rioul. "A historical perspective on Schützenberger-Pinsker inequalities (extended version)". In: *Information Geometry* (May 2024). ISSN: 2511-249X. DOI: 10.1007/s41884-024-00138-z. URL: http://dx.doi.org/10.1007/s41884-024-00138-z.

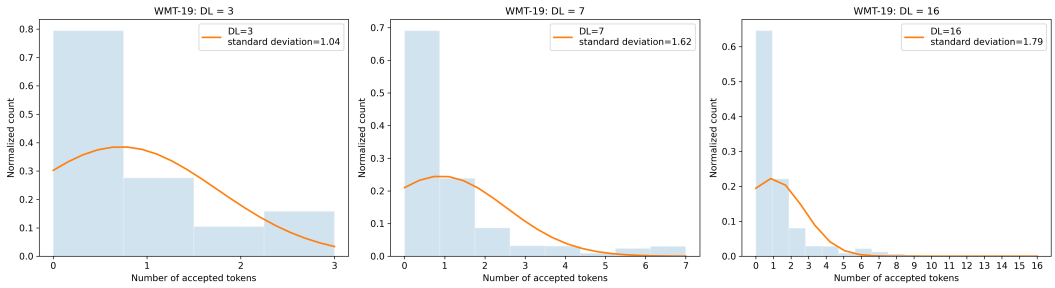
[22] Kushal Arora et al. *The Stable Entropy Hypothesis and Entropy-Aware Decoding: An Analysis and Algorithm for Robust Natural Language Generation*. 2023. arXiv: 2302.06784 [cs.CL]. URL: https://arxiv.org/abs/2302.06784.

A Acceptance rate variance for standard speculative decoding systems

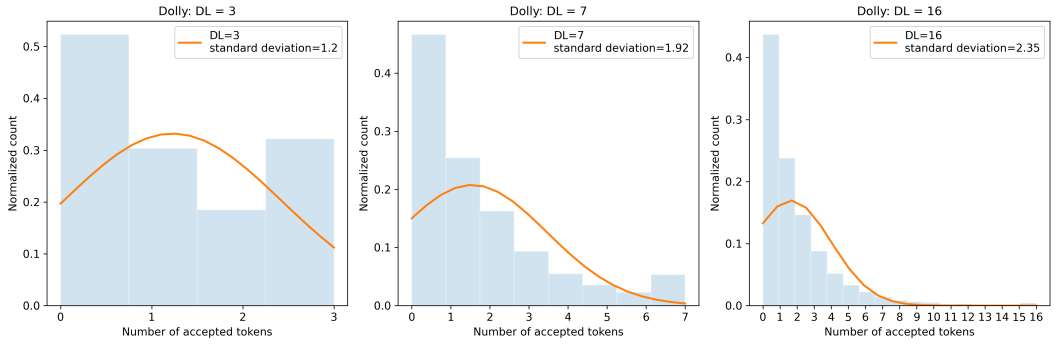
We see that across tasks like summarization (CNN-DM) in Figure 7a, translation (WMT-19) in Figure 7b, and creative writing (Dolly-15k) in Figure 7c, standard speculative decoding systems display a large variance in the number of tokens accepted per drafting round. This is observed across draft lengths 3, 7, and 16. This effect is particularly pronounced in the CNN-DM and Dolly-15k datasets, in which we observe standard deviations of ~ 2 tokens even within a maximum draft length of only 7 (i.e., 29% standard deviation), opening up room for significant optimization. Experiments are conducted with target model chosen as Llama2-7B and a 115M draft model finetuned via direct alignment with the target model distribution [18].



(a) Dataset: CNN-DM



(b) Dataset: WMT-19



(c) Dataset: Dolly-15k

Figure 7: Across datasets and draft lengths, the number of accepted tokens across drafting rounds displays a high variance, leading to under or over-utilization of the draft model in static draft length methods. Target model = Llama2-7B, draft model = Llama2-Drafter-115M, sampling temperature = 1.0.

B Derivation of entropy-based draft-stopping criteria

Let $p_{DM}(x)$ and $p_{TM}(x)$ be the currently observed probability distributions of the draft and target model given some prefix $x_{<t}$. Following [1], the acceptance probability, β of a token drafted from $p_{DM}(x)$ defined via rejection sampling is

$$\beta = \mathbb{E}_{x \sim p_{DM}(x)} \begin{cases} 1 & \text{if } p_{DM}(x) \leq p_{TM}(x) \\ \frac{p_{TM}(x)}{p_{DM}(x)} & \text{if } p_{DM}(x) > p_{TM}(x) \end{cases}$$

On discrete domains, the total variation distance between these distributions is defined in the following manner [20] :

$$TVD(p_{DM} \| p_{TM}) = \frac{1}{2} \sum_{x} |p_{TM}(x) - p_{DM}(x)|$$

We see in [1] that $TVD(DM \| TM)$ is related to the acceptance probability, β as

$$\beta = 1 - TVD(p_{DM} \| p_{TM})$$
$$\implies TVD(p_{DM} \| p_{TM}) = 1 - \beta$$

Moreover, by Pinsker's inequality [21], we may relate the total variation distance to the Kullback-Liebler divergence, $KLD(p_{DM} \| p_{TM})$

$$TVD(p_{DM} \| p_{TM}) \leq \sqrt{\frac{1}{2} KLD(p_{DM} \| p_{TM})}$$
$$\implies 1 - \beta \leq \sqrt{\frac{1}{2} KLD(p_{DM} \| p_{TM})}$$

Giving us,

$$1 - \sqrt{\frac{1}{2} KLD(p_{DM} \| p_{TM})} \leq \beta$$

Further, $KLD(p_{DM} \| p_{TM})$ relates to the cross-entropy $CE(p_{DM}, p_{TM})$ via

$$KLD(p_{DM} \| p_{TM}) = CE(p_{DM}, p_{TM}) - H_{DM}$$

where H_{DM} is the entropy of the draft model distribution.

Now, let us note, while drafting, we do not yet have access to the target model distribution $p_{TM}(x)$. We do know, however, that since $KLD \geq 0$, we have that

$$CE(p_{DM}, p_{TM}) \geq H_{DM}$$

In this work, we choose a linear approximation of $CE(p_{DM}, p_{TM})$ via a positive factor $\gamma' \in (0, 1)$

$$CE(p_{DM}, p_{TM}) = (1 + \gamma') H_{DM}$$

This is motivated by the observation that in LLM systems, most of the variation seen in the cross-entropy between the draft and target model occurs due to the high entropy of the draft model. LLMs suitable to be target models follow the stable entropy hypothesis [22] with reasonable generations lying in a narrow entropy band.

Substituting this approximation, we have that

$$
\begin{aligned}
1 - \sqrt{\frac{1}{2} KLD\left(p_{DM} \| p_{TM}\right)} &\leq \beta \\
1 - \sqrt{\frac{1}{2} KLD\left(p_{DM} \| p_{TM}\right)} &= 1 - \sqrt{\frac{1}{2}\left(CE(p_{DM}, p_{TM}) - H_{DM}\right)} \\
&\approx 1 - \sqrt{\frac{1}{2}\left(\left(1 + \gamma'\right) H_{DM} - H_{DM}\right)} \\
&= 1 - \sqrt{\gamma H_{DM}}
\end{aligned}
$$

Thus, we see that the value $1 - \sqrt{\gamma H_{DM}}$ acts as an approximate *lower bound* on the acceptance probability β. By stopping the drafting of a new token if our lower-bound estimate falls below a threshold λ, we attempt to ensure that the acceptance probability of the potential new token will be *greater* than this threshold. If the acceptance probability does not meet our threshold, we choose not to draft the next token.

Thus, a draft-stopping criteria

$$1 - \sqrt{\gamma H_{DM}(x)} < \lambda$$

implies that if drafting continues because $1 - \sqrt{\gamma H_{DM}(x)} \geq \lambda$, then we have an approximate lower bound on the acceptance probability of the drafted token via λ, i.e.,

$$\beta \geq \lambda$$

B.1 Computational cost

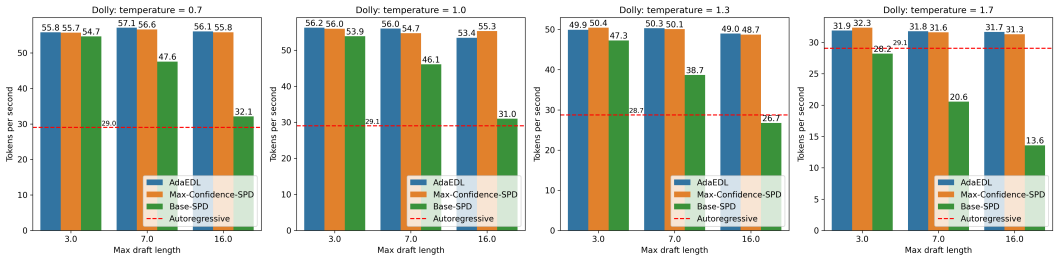
The cost of computing entropy is $O(N)$ on a single thread where N is the size of the vocabulary. That said, this operation is highly parallelizable since the N operations are independent. Thus, the overhead of AdaEDL is at most $O(N)$, but may be significantly reduced if implemented efficiently.

B.2 Impact of hardware chosen

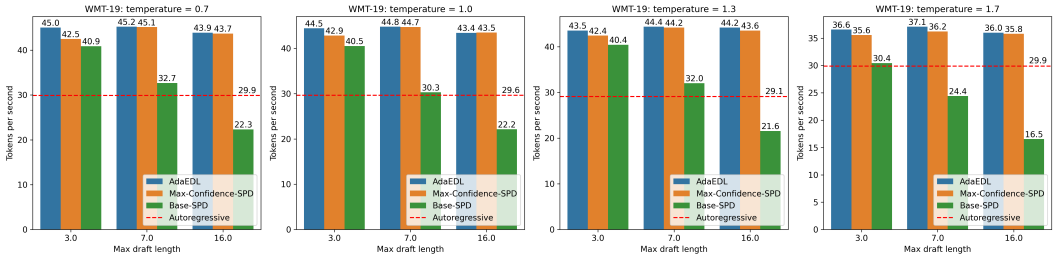
An additional consideration is that the speedup achievable by a speculative decoding system depends on the cost of running the draft model, which depends on its size and the nature of the hardware it runs on. For example, a larger draft model may have a higher acceptance rate, but is also more expensive to run. An ideal system is one that balances these factors, taking into account the expected acceptance rate and computational cost incurred on a particular hardware. Future work that studies draft model cost across various processors would be valuable to designing such a system.

C Effect of target model sampling temperature

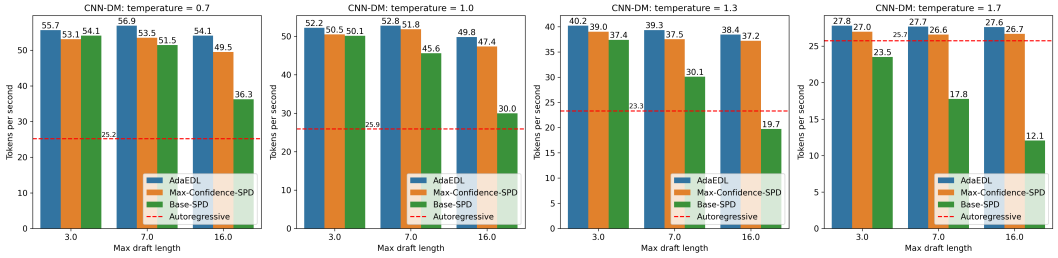
AdaEDL consistently outperforms the other 3 decoding methods across **sampling temperatures**. This trend is reflected across maximum draft lengths 3, 7, 16 and across datasets as seen for the Dolly-15k dataset (Figure 8a), the WMT-19 dataset (Figure 8b), and the CNN-DM dataset (Figure 8c).



(a) Dataset = Dolly-15k (creative writing)



(b) Dataset = WMT-19 (German-English translation)

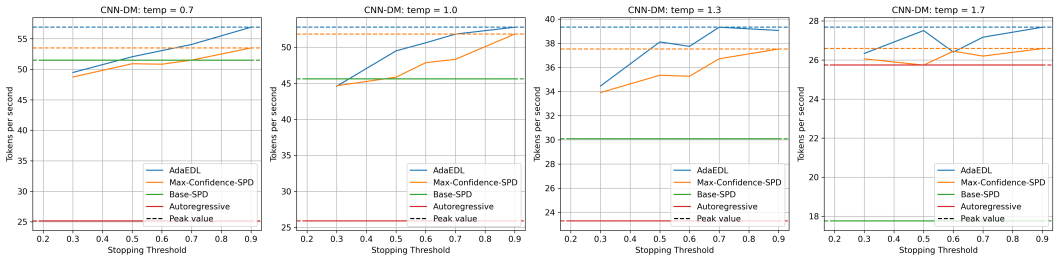


(c) Dataset = CNN-DM (summarization)

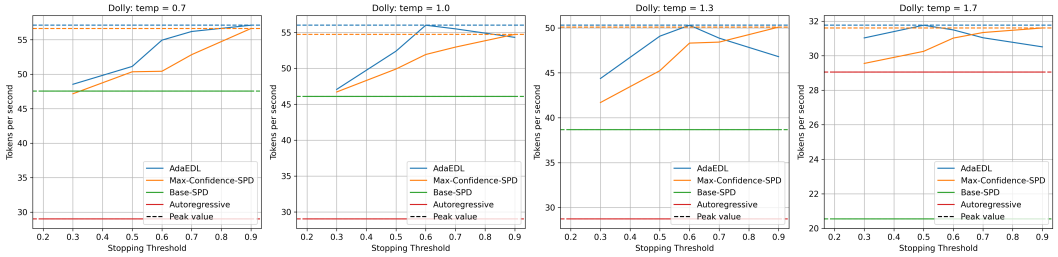
Figure 8: AdaEDL boosts token rate even in high temperature scenarios where Base-SPD is insufficient. Target Model = Llama2-7B, Draft Model = Llama2-Drafter-115M.

D Sensitivity to stopping thresholds

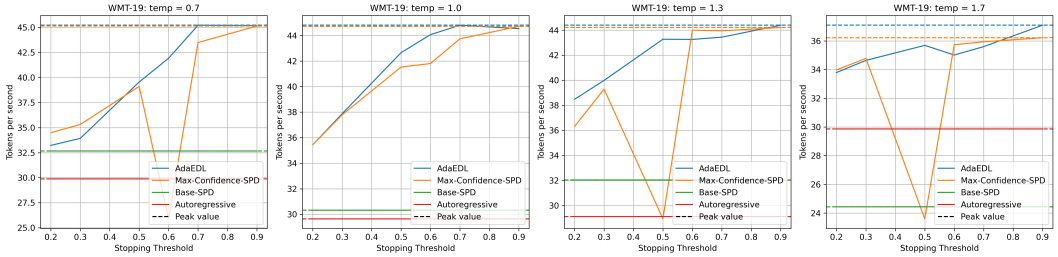
We see that AdaEDL is less sensitive to the choice of the **stopping threshold** λ, outperforming Max-Confidence-SPD even when suboptimal λ is chosen. This is reflected across temperatures and datasets as seen in Figures 9a, 9b, 9c.



(a) Dataset = CNN-DM (summarization)



(b) Dataset = Dolly-15k (creative writing)



(c) Dataset = WMT-19 (German-English translation)

Figure 9: AdaEDL maintains a margin over other methods even for sub-optimal stopping threshold choices. Max DL = 7, TM = Llama2-7B, DM = Llama2-Drafter-115M.