Speculative Contrastive Decoding

Anonymous ACL submission

Abstract

Large language models (LLMs) exhibit exceptional performance in language tasks, yet their auto-regressive inference is limited due to high computational requirements and is sub-optimal due to the exposure bias. Inspired by speculative decoding and contrastive decoding, we introduce Speculative Contrastive Decoding (SCD), a straightforward yet powerful decoding approach that leverages predictions from smaller language models (LMs) to achieve both decoding acceleration and quality improvement. Extensive evaluations and analyses on four diverse language tasks demonstrate the effectiveness of SCD, showing that decoding efficiency and quality can compatibly benefit from one smaller LM.

1 Introduction

Large language models (LLMs) have advanced the versatility and proficiency in approaching real-world natural language tasks such as general instruction following (Ouyang et al., 2022; Taori et al., 2023; Lu et al., 2023) and reasoning (Cobbe et al., 2021; Wei et al., 2023; Yuan et al., 2023). Most existing LLMs (Brown et al. (2020); Touvron et al. (2023); Bai et al. (2023),*inter alia*) are built on decoder-only Transformers. Due to the auto-regressive nature during inference, the runtime of decoding inference can be excessive on general computation infrastructure, and the generation quality can be sub-optimal due to the exposure bias (Arora et al., 2022). Improving decoding inference has been the spotlight of the research community in language generation (Vijayakumar et al., 2018; Holtzman et al., 2020; Su et al., 2022).

As for decoding acceleration, one prominent method named speculative decoding (Leviathan et al., 2022; Chen et al., 2023) has been proposed and leverages relatively smaller language models (LMs) to predict several successive token generations of target LLMs. The LLMs only require one-time forward computation for checking the validity of predictions from the smaller LMs. The decoding method maintains the target LLMs' token distributions and accelerates more when smaller LMs can accurately predict the potential target LLMs' generations.

As for the generation quality, contrastive decoding has been recently proposed (Li et al., 2023a). Contrastive decoding assumes that conjugated smaller LMs may present higher systematic tendencies to generate erroneous tokens than the larger ones, and the method seeks to eliminate such systematic error by contrasting the token distribution between smaller LMs and larger LMs. From either inference acceleration or quality improvement, these works have demonstrated a promising direction by integrating smaller LMs during auto-regressive generation.

Inspired by both speculative and contrastive decoding, we propose Speculative Contrastive Decoding (SCD), which exploits a single smaller LM for decoding improvement in speed and quality en bloc. Comprehensive evaluations of four diverse tasks show that SCD can achieve similar acceleration factors of speculative decoding while maintaining the quality improvement from contrastive decoding. By further analyzing the token distributions of the smaller and larger LMs in SCD, we show the inherent compatibility of decoding acceleration and quality improvement. The contributions of this paper can be summarized as follows:

- We propose Speculative Contrastive Decoding for efficacious LLM inference.
- Comprehensive experiments and analysis illustrate the compatibility of speculative and contrastive decoding on 4 diverse tasks.

2 Related Works

In terms of inference acceleration, recent research has been devoted to developing various efficient decoding methods (Yao et al., 2022; Kwon et al.,

2023; Cai et al., 2023). Speculative decoding Leviathan et al. (2022); Chen et al. (2023); Kim et al. (2023) is one of these recent works and utilizes smaller models for acceleration. Miao et al. (2023); Spector and Re (2023) propose to organize predictions from small LMs into tree structures to accelerate speculative decoding further. In terms of inference quality, rich research has been suggested (Vijayakumar et al., 2018; Holtzman et al., 2020; Su et al., 2022; Su and Xu, 2022; Finlayson et al., 2023) and contrastive decoding achieves better decoding qualities by similarly integrating smaller LMs and devise contrastive token distributions (Li et al., 2023a; O'Brien and Lewis, 2023). It can further be adjusted to other variants such as the token distribution contrasting between model layers (Chuang et al., 2023) or different inputs (Yona et al., 2023). SCD draws inspiration from these works and benefits both decoding speed and quality by incorporating smaller LMs into generation.

3 Preliminaries

We follow the terminology in Li et al. (2023a), and term the target larger LMs as the expert LMs while the smaller LMs as the amateur LMs denoted as $\mathcal{M}_e$ and $\mathcal{M}_a$ respectively.

3.1 Contrastive Decoding

The intrinsic rationale of contrastive decoding (CD) is that amateur LMs have stronger systematic undesirable tendencies to produce undesirable patterns (e.g., hallucination) than expert LMs. By contrasting the token distributions between expert and amateur LMs, such tendencies can be alleviated. There have been successively proposed two versions of contrastive decoding by Li et al. (2023a) and O'Brien and Lewis (2023), which we term as *Original* contrastive decoding and *Improved* contrastive decoding. The final contrastive logit scores for the original contrastive decoding $s_{\text{ori}}(x_i|x_{<i})$ and the improved contrastive decoding $s_{\text{imp}}(x_i|x_{<i})$ are respectively:

$$s_{\text{ori}}(x_i|x_{<i}) = \begin{cases} \log P_{\mathcal{M}_e}(x_i|x_{<i}) - \log P_{\mathcal{M}_a}(x_i|x_{<i}), & x_i \in \mathcal{V}^{\alpha}_{\text{ori},i} \\ -\infty, & x_i \notin \mathcal{V}^{\alpha}_{\text{ori},i} \end{cases}$$

$$s_{\text{imp}}(x_i|x_{<i}) = \begin{cases} (1+\beta) Y_{\mathcal{M}_e}(x_i|x_{<i}) - \beta Y_{\mathcal{M}_a}(x_i|x_{<i}), & x_i \in \mathcal{V}^{\alpha}_{\text{imp},i} \\ -\infty, & x_i \notin \mathcal{V}^{\alpha}_{\text{imp},i} \end{cases}$$

where $P_{\cdot}$ and $Y_{\cdot}$ are respectively the token probability and logit generated from LMs. $\mathcal{V}^{\alpha}_{\cdot,i}$ denotes the adaptive plausibility constraint that dynamically restricts the logits from producing the erroneous modes. The adaptive plausibility constraints are calculated as

$$\mathcal{V}^{\alpha}_{\text{ori},i} = \left\{ w | P_{\mathcal{M}_e}(w|x_{<i}) > \alpha \max_{w \in \mathcal{V}} P_{\mathcal{M}_e}(w|x_{<i}) \right\},$$

$$\mathcal{V}^{\alpha}_{\text{imp},i} = \left\{ w | Y_{\mathcal{M}_e}(w|x_{<i}) > \log \alpha + \max_{w \in \mathcal{V}} Y_{\mathcal{M}_e}(w|x_{<i}) \right\}.$$

A token is generated from the contrastive token distribution $P^{\tau}_n(x_i) = \text{softmax}_{\tau}\left(s_n(x_i|x_{<i})\right)$, $n \in \{\text{ori}, \text{imp}\}$, where τ represents the softmax temperature that determines the smoothness of the contrastive token distribution.

Algorithm 1: Speculative Contrastive Decoding

Data: $\mathcal{M}_e$, $\mathcal{M}_a$, input prefix x_{inp}
Result: $[x_{\text{inp}}, x_1, .., x_{\text{k}}]$
for i *from* 1 *to* γ **do**
  $x_i \sim P_{\mathcal{M}_a}(x_i) = \mathcal{M}_a(x_i|x_{\text{inp}}, x_{<i})$;
$P_{\mathcal{M}_e}(x_1), .., P_{\mathcal{M}_e}(x_{\gamma+1}) = \mathcal{M}_e(x_1, .., x_{\gamma}|x_{\text{inp}})$;
Calculate $P_n(x_1), .., P_n(x_{\gamma})$ following Section §3.1;
$r_1, .., r_{\gamma}$ i.i.d sampled from Uniform$(0, 1)$;
$k = \min\left(\{i | r_i > \frac{P_n(x_i)}{P_{\mathcal{M}_a}(x_i)}\} \cup \{\gamma + 1\}\right)$;
if $k \leq \gamma$ **then**
  $P_k(x_k) = \text{norm}(\max(0, P_n(x_k) - P_{\mathcal{M}_a}(x_k))$;
  Resample $x_k \sim P_k(x_k)$;
else
  $P_{\mathcal{M}_a}(x_{\gamma+1}) = \mathcal{M}_a(x_{\gamma+1}|x_{\text{inp}}, x_1, .., x_{\gamma})$;
  Calculate $P_n(x_{\gamma+1})$ following Section §3.1;
  $x_{\gamma+1} \sim P_n(x_{\gamma+1})$;

3.2 Speculative Decoding

Instead of requiring one forward computation of $\mathcal{M}_e$ for each token in vanilla decoding, speculative decoding (SD) utilizes $\mathcal{M}_a$ to primarily generate γ tokens at each iteration then $\mathcal{M}_e$ makes one forward computation to check the validity of the γ tokens. If $\mathcal{M}_e$ accepts all the γ tokens, it finishes the iteration with an additional generated token, resulting in $\gamma + 1$ tokens generated. Otherwise, if $\mathcal{M}_e$ rejects a token at r, the token is re-sampled according to $\mathcal{M}_e$ to substitute the rejected token; hence the iteration finishes with r tokens generated. With only one-time forward computation of $\mathcal{M}_e$, multiple tokens are generated at each iteration. When the ratio between the runtime required of $\mathcal{M}_a$ and $\mathcal{M}_e$ (the cost coefficient c, Leviathan et al. (2022)) is low and the token acceptance rate is high, there will present a notable acceleration.

4 Speculative Contrastive Decoding

Speculative decoding leverages smaller $\mathcal{M}_a$ only for generation acceleration, while not making the

best of the token distributions from $\mathcal{M}_a$. It is natural to simultaneously apply the contrastive token distribution, and with negligible computational overhead, the generation quality and efficiency can benefit from integrating speculative and contrastive decoding. Therefore, we propose Speculative Contrastive Decoding (SCD).

Concretely, at each iteration, γ tokens are generated from the amateur model $\mathcal{M}_a$. When checking the validity of the tokens, the target distribution becomes $P_n^\tau, n \in \{\text{ori}, \text{imp}\}$ from contrastive distribution instead of $P_{\mathcal{M}_e}$ in speculative decoding. For a token x in the $\mathcal{M}_a$-generated tokens, it is rejected with probability $1 - \frac{P_n^\tau(x)}{P_{\mathcal{M}_a}(x)}$ and then a new token in place of x is re-sampled from $\text{norm}(\max(0, P_n^\tau(x) - P_{\mathcal{M}_a}(x))$, where $\text{norm}\left(f(x)\right) = f(x)/\sum_x f(x), \text{s.t.} f(x) \geq 0$. If all the $\mathcal{M}_a$-generated tokens are accepted, then an additional token is sampled from P_n^τ.

The sampling procedure of SCD is similar to the original speculative decoding in Leviathan et al. (2022); Chen et al. (2023). However, it is worth noticing that in our SCD, when all the $\mathcal{M}_a$-generated tokens are accepted, we require an additional forward computation from $\mathcal{M}_a$ to acquire its last token logit for calculating the contrastive distribution P_n^τ at that iteration, while in speculative decoding, the additional token is sampled directly from $\mathcal{M}_e$. This computational overhead is negligible when c is small. We detailed the algorithm of our SCD in Algorithm Alg. 1. The difference from the original speculative decoding is highlighted in blue.

5 Experiment

Experiment Setting. We evaluate SCD and other baselines on four benchmarks: **WikiText** (Merity et al., 2016), **HumanEval** (Chen et al., 2021), **AlpacaEval** (Li et al., 2023b), and **GSM8k** (Cobbe et al., 2021). The four benchmarks span diverse language tasks of open-ended generation, code generation, human alignment, and mathematical reasoning respectively. For WikiText, we use the pre-trained $\text{Llama2}_{7\text{B}}$ and $\text{Llama2}_{70\text{B}}$ (Touvron et al., 2023) as $\mathcal{M}_a$ and $\mathcal{M}_e$ and follow Li et al. (2023a) to use diversity, MAUVE (Pillutla et al., 2021) and coherence as evaluation metrics. For HumanEval, we use the pre-trained $\text{Llama2}_{7\text{B}}$ and $\text{Llama2}_{70\text{B}}$ and assess the 1-round pass rate. For AlpacaEval, we use human-aligned $\text{Llama2chat}_{7\text{B}}$ and $\text{Llama2chat}_{70\text{B}}$ and report win-rates over *text-davinci-003* judged by GPT-4. For GSM8k, we use fine-tuned $\text{Llama2}_{7\text{B}}$ and $\text{Llama2}_{70\text{B}}$ on its training set and report the accuracy of the test-set results. We set $\gamma = 4$ across all experiments and set the temperature τ to 0.7 for WikiText and AlpacaEval and 0.001 for GSM8k and HumanEval. We leave the detailed experiment settings to Appx. §A.

	WikiText			A.Eval	GSM8k	H.Eval
	Div.	MAU.	Coh.	Score	Acc.	Pass@1
$\mathcal{M}_a$	$0.69_{.00}$	$0.88_{.01}$	$0.76_{.00}$	$88.79_{1.1}$	$41.77_{.00}$	$11.59_{.0}$
$\mathcal{M}_e$	$0.75_{.00}$	$0.88_{.01}$	$0.75_{.00}$	$94.66_{.79}$	$64.19_{.04}$	$28.66_{.0}$
SD	$0.75_{.00}$	$0.90_{.01}$	$0.75_{.01}$	$94.28_{.83}$	$64.27_{.07}$	$28.66_{.0}$
CD_{ori}	$0.91_{.00}$	$0.95_{.00}$	$0.73_{.00}$	$94.56_{.82}$	$64.42_{.03}$	$37.20_{.0}$
SCD_{ori}	$0.91_{.00}$	$0.94_{.00}$	$0.72_{.01}$	$94.91_{.78}$	$64.44_{.06}$	$33.54_{.0}$
E.A._{ori}		$\times 1.78$		$\times 2.92$	$\times 3.32$	$\times 3.01$
CD_{imp}	$0.73_{.01}$	$0.90_{.01}$	$0.74_{.00}$	$94.78_{.79}$	$64.91_{.01}$	$37.20_{.0}$
SCD_{imp}	$0.73_{.00}$	$0.91_{.01}$	$0.74_{.00}$	$95.03_{.77}$	$64.90_{.02}$	$33.54_{.0}$
E.A._{imp}		$\times 2.10$		$\times 2.95$	$\times 3.32$	$\times 3.18$

Table 1: Main results of SCD. H.Eval, and A.Eval are shorts for HumanEval and AlpacaEval. MAU. and Coh. are shorts for MAUVE and coherence. E.A. presents the expected acceleration under $c = 0.05$. The standard errors under 3 repetitions for each result are marked in subscripts. The best choices of α and β for (S)CD are left to Appx. §A.3.

Quality Results. As shown in Tab. 1, original and improved SCD and CD demonstrate significant improvement over $\mathcal{M}_e$ in GSM8k and HumanEval. On WikiText, only original CD and SCD outperform $\mathcal{M}_e$ in terms of diversity with $+0.16$ and MAUVE with $+0.06$. There is no obvious improvement in Coherence. On AlpacaEval, although both versions of SCD and CD show better results than $\mathcal{M}_e$, such improvement is not significant due to the high variance of GPT4-as-a-judge. We can see that different versions of SCD suggest different levels of improvement. Original SCD performs better on WikiText and HumanEval while inferior on GSM8k to improved SCD. Results across four benchmarks show SCD can benefit various LLMs on diverse language tasks, maintaining the same generation quality improvement as CD.

Acceleration. To demonstrate the inference acceleration of SCD, we primarily provide the expected acceleration factor of SCD theoretically with respect to the number of $\mathcal{M}_a$ token predictions per iteration γ, the acceptance rate λ, and the cost coefficient c, which proof is left to Appx. §B.

Theorem 5.1. *The expected acceleration factor in*

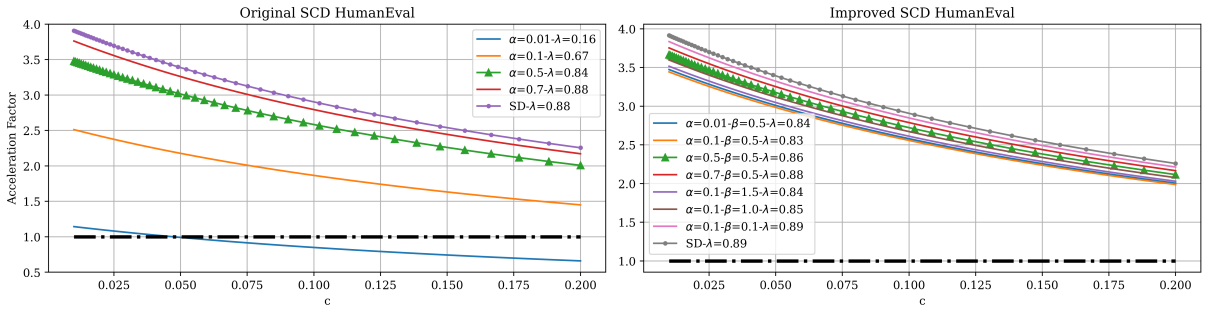

Figure 1: Hyper-parameter analysis on expected acceleration factors regarding empirical acceptance rate λ. The best hyper-parameter settings as in Tab. 1 are the lines marked with triangles.

Figure 2: The averaged token distribution entropy with error bars of rejected and accepted tokens in SCD.

decoding runtime is $\frac{1-\lambda^{\gamma+1}}{(1-\lambda)(1+c\gamma+c\lambda^{\gamma})}$.

In Tab. 1, consistent acceleration is presented across different benchmarks. We further visualize the expected acceleration factor of SCD in Fig. 1 according to the empirical acceptance rates λ in HumanEval with different hyper-parameter settings. According to Theorem 5.1, the acceleration factors are depicted against the cost coefficient c, which is usually of small values representing the ratio of runtime required of $\mathcal{M}_a$ and $\mathcal{M}_e$ and depends on the infrastructures (e.g., GPU) that serve the LLMs. We can see that the acceptance rates hence the corresponding acceleration factors of original SCD are more sensitive to hyper-parameters compared to improved SCD. With proper hyper-parameters, SCD can achieve similar acceleration to the speculative decoding (dotted lines), which indicates the negligible speed trade-off to incorporate the contrastive token distributions. Results on GSM8k are listed in Appx. §D presenting similar patterns.

6 Analysis

Compatibility. Results presented in §5 show SCD can combine the benefits of CD and SD. We delve deep into the reasons for such compatibility. We calculate the average entropy of token probabilities from $\mathcal{M}_a$ and $\mathcal{M}_e$ regarding the accepted and rejected tokens in SCD. As shown in Fig. 2, token distribution entropy from both $\mathcal{M}_a$ and $\mathcal{M}_e$ of accepted tokens is significantly higher than that of rejected tokens. The phenomenon suggests

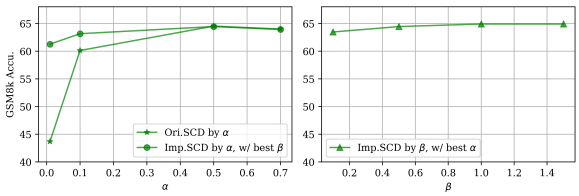

Figure 3: Performance sensitivity regarding α and β.

SCD enjoys acceleration from accepting easy tokens of lower entropy while benefiting from contrastive token distribution by rejecting hard tokens of higher entropy. We also present a case study from GSM8k in Appx. §C to demonstrate such compatibility.

Sensitivity. Through Fig. 3, we show how performances fluctuate with respect to the hyper-parameter α and β. We can see that improved SCD is less sensitive to both α and β on GSM8k compared to the original SCD. This is possibly due to the better flexibility of manipulating logits than probabilities. Results on HumanEval are listed in Appx. §D presenting similar phenomenons.

7 Conclusion

In this paper, we propose speculative contrastive decoding, a decoding strategy that naturally integrates small amateur LMs for inference acceleration and quality improvement of LLMs. Extensive experiments show the effectiveness of SCD and our delve-deep analysis also explains the compatibility through the scope of token distribution entropy. Our method can be easily deployed to improve the real-world serving of LLMs.

Limitation

In our experiments, we provide the expected acceleration factors of SCD on four benchmarks calculated according to the empirical token acceptance rates λ and selected cost coefficients c. The empirical acceleration factor is highly correlated to the actual infrastructures that serve both the larger LMs and the smaller LMs. To compensate for this demonstration limitation and better demonstrate the acceleration performance, we visualize the expected acceleration factor by spanning across a range of c in Fig. 1. This is a common limitation of deploying speculative decoding in the real-world LLM serving. For example, the runtime of switching between the forward computation of $\mathcal{M}_a$ and $\mathcal{M}_e$ would be non-negligible without properly optimized infrastructures, causing a relatively large c hence potentially resulting in deceleration even with high acceptance rates.

Broader Impact

Although LLMs have demonstrated exceptional performance and been helpful real-world assistants recently, the massive computational demands of LLMs forbid most users including potential researchers from local deployments, who generally alter to use APIs from LLM servings. Therefore, effective methods, including our SCD, to improve the speed and quality from the perspective of decoding inference have much potential to advance LLM-based services.

References

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Cheung. 2022. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 700–710, Dublin, Ireland. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

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. 2023. Accelerating large language model decoding with speculative sampling.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

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

Matthew Finlayson, John Hewitt, Alexander Koller, Swabha Swayamdipta, and Ashish Sabharwal. 2023. Closing the curious case of neural text degeneration.

Mingqi Gao and Xiaojun Wan. 2022. DialSummEval: Revisiting summarization evaluation for dialogues. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5693–5709, Seattle, United States. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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*.

Sehoon Kim, Karttikeya Mangalam, Suhong Moon, Jitendra Malik, Michael W. Mahoney, Amir Gholami, and Kurt Keutzer. 2023. Speculative decoding with big little decoder.

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. 2022. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023a. 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)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

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*.

Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems*.

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

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation.

Yixuan Su and Jialu Xu. 2022. An empirical study on contrastive search and contrastive decoding for open-ended text generation. *arXiv preprint arXiv:2211.10797*.

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, 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. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search: Decoding diverse solutions from neural sequence models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183.

Gal Yona, Or Honovich, Itay Laish, and Roee Aharoni. 2023. Surfacing biases in large language models using contrastive input decoding. *arXiv preprint arXiv:2305.07378*.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models.

A Experiment Details

A.1 Benchmark Details

(1) **WikiText** (Merity et al., 2016) contains articles from Wikipedia. We follow the pre-processing scripts from Li et al. (2023a) and result in 1,733 samples. The generation starts with the first 32 tokens as prompts, and the max generation length is set to 256. We report diversity, MAUVE (Pillutla et al., 2021), and coherence as metrics, following Li et al. (2023a).

Diversity metrics assess the unique multi-grams in the completion generated from the LMs. Higher diversity scores indicate better lexical diversity in the completion. The diversity is calculated according to:

$$\text{Div.} = \prod_{n=2}^{4} \frac{|\text{Set(n-grams)}|}{|\text{n-grams}|}.$$

MAUVE is a metric proposed by Pillutla et al. (2021), which is empirically suggested to have better agreement with human annotations (Gao and Wan, 2022). Coherence evaluates the semantic correlation between the input prefix and the output generation via the similarity of embeddings. We use the sentence embeddings following SimCSE (Gao et al., 2021) and the coherence score is calculated as:

$$\frac{\text{emb}(x_{\text{prefix}}) \cdot \text{emb}(x_{\text{gen}})}{\|\text{emb}(x_{\text{prefix}})\| \|\text{emb}(x_{\text{gen}})\|}.$$

(2) **GSM8k** (Cobbe et al., 2021) contains training and evaluation sets of grade mathematical reasoning problems. We first fine-tune the $\text{Llama2}_{\text{7B}}$ and $\text{Llama2}_{\text{70B}}$ by 3 epochs to produce the amateur and expert LMs. We report the final accuracy of the test sets.

(3) **HumanEval** (Chen et al., 2021) measures coding correctness for synthesizing programs from 164 doc-strings. We report the 1-round pass rate (Pass@1).

(4) **AlpacaEval** (Li et al., 2023b) contains 805 samples from various evaluation sets to evaluate the alignment abilities of LLMs by comparing evaluated models with *text-davinci-003*. We report the win rate judged by GPT-4.

A.2 Configuration Details

We use $\text{Llama2}_{\text{7B}}$ as the amateur model while $\text{Llama2}_{\text{70B}}$ as the expert model on WikiText and HumanEval benchmarks to evaluate how SCD performs with pre-trained models. Then, we fine-tune $\text{Llama2}_{\text{7B}}$ and $\text{Llama2}_{\text{70B}}$ on the GSM8k training set to evaluate the SCD performance with supervised fine-tuning models on the mathematical reasoning task. We also apply $\text{Llama2chat}_{\text{7B}}$ and $\text{Llama2chat}_{\text{70B}}$ on AlpacaEval to assess LLMs for human alignment using SCD. We set the softmax temperature consistent to 0.7 on WikiText and AlpacaEval while 0.001 on other benchmarks. In SCD and SD, we always set the prediction temperature from the amateur LMs to 1.0 for fair comparison. All experiments are conducted on 2 A100 80G GPUs with KV cache implementation.

A.3 Hyper-parameter Details

We conduct grid searches regarding α and β for the best performance of CD and SCD. The best hyper-parameter settings for the results in Tab. 1 are listed in Tab. 2.

B Proof of Theorem Theorem 5.1

Theorem B.1. *The expected acceleration factor in decoding runtime is* $\frac{1-\lambda^{\gamma+1}}{(1-\lambda)(1+c\gamma+c\lambda^{\gamma})}$.

Proof. Similar to Theorem 3.8 in Leviathan et al. (2022), given the token acceptance rate λ and the runtime per forward computation step for $\mathcal{M}_e$ and $\mathcal{M}_a$ are T and cT. The total runtime required for each iteration is $T + c\gamma T + c\lambda^{\gamma}T$, where $\mathcal{M}_a$ requires γ generation steps and possibly one additional step forward computation if all γ tokens are accepted while $\mathcal{M}_a$ requires one forward computation for token validity checking. Following Equation (1) in Leviathan et al. (2022), the expected generated token number per iteration is $\frac{1-\lambda^{\gamma+1}}{1-\lambda}$. Therefore, the expected runtime needed of SCD is $\frac{1-\lambda}{1-\lambda^{\gamma+1}}(T+c\gamma T+c\lambda^{\gamma}T)$, hence the expected acceleration factor is $\frac{1-\lambda^{\gamma+1}}{(1-\lambda)(1+c\gamma+c\lambda^{\gamma})}$. $\square$

	WikiText		AlpacaEval		GSM8k		HumanEval	
	α	β	α	β	α	β	α	β
CD_{ori}	0.1	-	0.5	-	0.5	-	0.5	-
SCD_{ori}	0.1	-	0.5	-	0.5	-	0.5	-
CD_{ori}	0.1	0.5	0.5	0.5	0.5	1.0	0.5	0.5
SCD_{ori}	0.1	0.5	0.5	0.5	0.5	1.0	0.5	0.5

Table 2: The hyper-parameter settings for the results in Tab. 1

C Case Study

Case Study on GSM8k

Judy teaches 5 dance classes, every day, on the weekdays and 8 classes on Saturday. If each class has 15 students and she charges $15.00 per student, how much money does she make in 1 week?

Responses

[RED and BLUE tokens are generated by the expert LM after rejection and all acceptance, respectively. Other tokens are generated by the amateur LM.]

She teaches 5 classes every weekday so that's 5*5 = «5*5=25»25 classes
She teaches 25 weekday classes and 8 Saturday classes for a total of 25+8 = «25+8=33»33 classes
Each class has 15 students so she teaches 15*33 = «15*33=495»495 students in 1 week
She charges $15.00 per student and she teaches 495 students so she makes 15*495 = $«15*495=7425.00»7,425.00 in 1 week

In this case, we can see that the rejected and re-sampled tokens are usually the beginning of a sentence, numbers, operations, or named entities, which are generally informative tokens in the reasoning chain of thoughts. This also indicates that quality improvement originates from re-sampling informative tokens by contrastive token distribution while the acceleration comes from speculative prediction of the amateur LMs.

D Additional Results

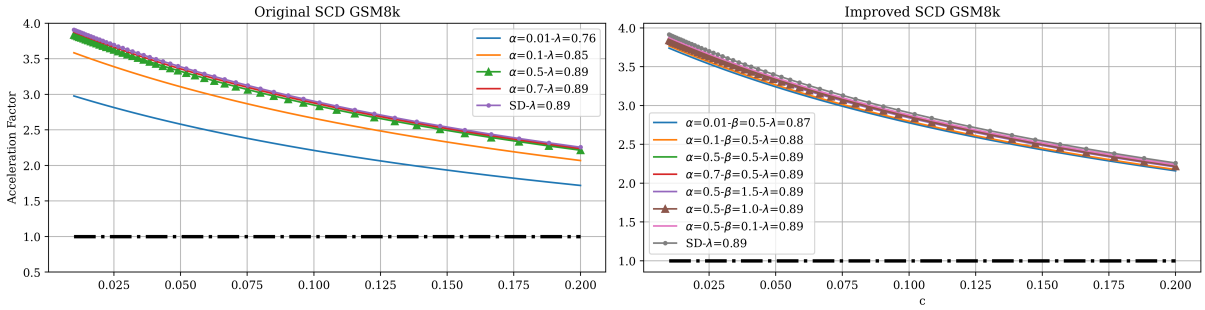

Figure 4: Hyper-parameter analysis on expected acceleration factors regarding empirical acceptance rate λ. The best hyper-parameter settings as in Tab. 1 are the lines marked with triangles.

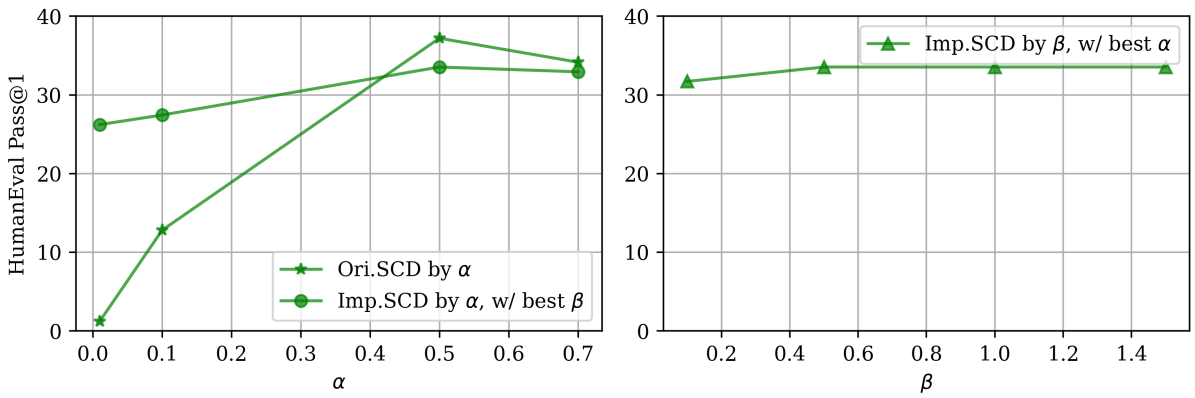

Figure 5: Performance sensitivity regarding α and β.