KOALA: Enhancing Speculative Decoding for LLM via Multi-Layer Draft Heads with Adversarial Learning

Kaiqi Zhang
Dalian University of Technology
Dalian, China
zhangkq@mail.dlut.edu.cn

Jing Zhao
Dalian University of Technology
Dalian, China
zhaoj9988@dlut.edu.cn

Rui Chen
Dalian University of Technology
Dalian, China
72117004@mail.dlut.edu.cn

***Abstract*—Large Language Models (LLMs) exhibit high inference latency due to their autoregressive decoding nature. While the draft head in speculative decoding mitigates this issue, its full potential remains unexplored. In this paper, we introduce KOALA (K-layer Optimized Adversarial Learning Architecture), an orthogonal approach to the draft head. By transforming the conventional single-layer draft head into a multi-layer architecture and incorporating adversarial learning into the traditional supervised training, KOALA significantly improves the accuracy of the draft head in predicting subsequent tokens, thus more closely mirroring the functionality of LLMs. Although this improvement comes at the cost of slightly increased drafting overhead, KOALA substantially unlocks the draft head's potential, greatly enhancing speculative decoding. We conducted comprehensive evaluations of KOALA, including both autoregressive and non-autoregressive draft heads across various tasks, demonstrating a latency speedup ratio improvement of 0.24x-0.41x, which is 10.57%-14.09% faster than the original draft heads.**

***Index Terms*—Large Language Models (LLMs), Speculative decoding, Draft head, Adversarial learning**

I. Introduction

Large Language Models (LLMs), such as GPT-4 [1], Llama 2 [2], and PaLM 2 [3], demonstrate exceptional performance across various tasks. Due to their inherent autoregressive decoding nature, accelerating LLM inference has become a crucial research objective. Speculative decoding [4], [5], utilizing a draft model, enhances the efficiency of target LLM inference through a *draft-then-verify* paradigm. In each iteration of speculative decoding, the draft model initially predicts multiple subsequent tokens, which are then concurrently verified by the target LLM for acceptable continuations.

Speculative decoding hinges on finding a draft model that closely mirrors the target LLM's functionality while achieving faster inference. Initial approaches employed *independent drafting*, wherein a smaller, separate LM (e.g., T5-small) accelerates inference for a larger LM (e.g., T5-XXL). However, LMs from disparate series frequently exhibit incompatible implementation details, hindering interoperability. Moreover, the high costs of training a dedicated LM for speculative decoding constrain the practicality of independent drafting. Recent advancements introduce *self-drafting* methods, which enhance LLM inference speed without relying on separate draft models. Numerous self-drafting techniques design lightweight draft models called *draft heads*, leveraging the semantically rich hidden states of the target LLM. Draft heads can be classified into two categories based on their decoding approach: *non-autoregressive* and *autoregressive*. Medusa [6] and EAGLE [7] are representative works in these respective domains.

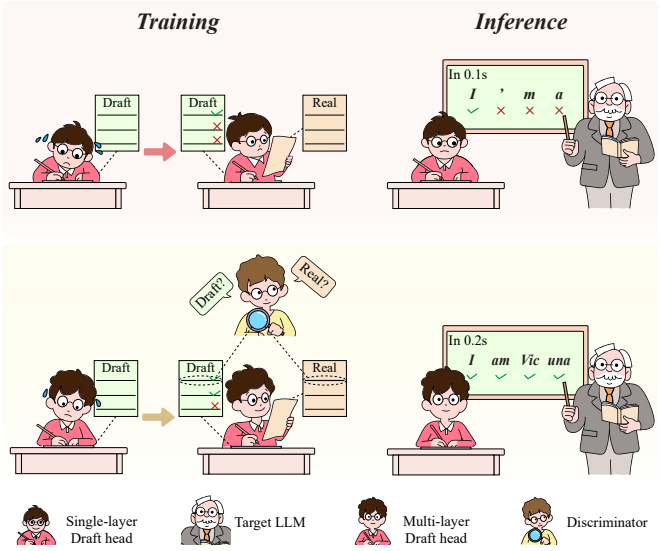

Fig. 1: Comparison between the traditional draft head (upper panel) and the KOALA-optimized draft head (lower panel). KOALA expands the conventional single-layer structure to a multi-layer architecture and incorporates adversarial learning into traditional supervised training. While KOALA slightly increases drafting overhead, it substantially enhances speculative decoding efficiency by improving the draft head's accuracy in predicting subsequent tokens.

Although draft heads achieve significant acceleration, several limitations persist: 1) Current draft heads employ a single-layer architecture, enabling rapid token prediction but resulting in a substantial performance gap compared to target LLMs due to parameter count disparity. This gap impedes effective collaboration between draft heads and target LLMs, limiting

their potential. 2) Current draft head training methods rely on supervised learning, which only captures superficial input-output mappings. This approach inadequately enables draft heads to capture the underlying process for generating tokens consistent with the target LLM's output distribution, limiting their predictive accuracy.

To address these limitations and unlock the potential of draft heads, we introduce KOALA, an orthogonal technique for draft head optimization, as illustrated in Figure 1. 1) We propose a *multi-layer* draft head structure to mitigate the performance gap with target LLMs caused by parameter disparities. In contrast to a single-layer design, this multi-layer architecture enables draft heads to more closely mirror target LLMs' functionality, enhancing overall collaboration. 2) We introduce a novel draft head training method that incorporates *adversarial learning* into traditional supervised training. By leveraging the dynamic game mechanism between draft heads and discriminators, this approach encourages draft heads to better capture intricate token generation details in target LLMs, significantly improving prediction accuracy. KOALA increases the number of tokens generated per draft-then-verify cycle, reducing the number of required algorithm iterations and enhancing speculative decoding efficiency. Notably, although the multi-layer structure slightly increases the draft overhead, it significantly accelerates the LLMs inference.

Our contributions can be summarized as follows:

- We introduce KOALA, an orthogonal approach to improving draft head prediction accuracy that enhances speculative decoding efficiency. Specifically, KOALA includes two key innovations: expanding the conventional single-layer draft head into a multi-layer architecture and incorporating adversarial learning into traditional supervised training.
- We evaluated KOALA on the MT-bench using Medusa and EAGLE to represent non-autoregressive and autoregressive draft heads, respectively, with Vicuna models (7B, 13B, 33B) as target LLMs. Experimental results demonstrate that KOALA achieves a 0.24x-0.41x improvement in latency speedup ratio, which is 10.57%-14.09% faster than the original draft heads.

II. KOALA

KOALA optimizes the draft head in speculative decoding through its distinct structure and training process. To demonstrate KOALA, we employed Medusa and EAGLE as representatives of non-autoregressive and autoregressive draft heads, respectively.

A. Multi-Layer Draft Head

To reduce the performance gap between the draft head and the target LLM, KOALA transformed the single-layer draft head into a multi-layer structure, as illustrated in Figure 2.

The traditional Medusa Head comprises a Residual Block (ResBlock) followed by a Linear layer. The ResBlock predicts features of subsequent tokens, while the Linear layer maps these features to the vocabulary size. KOALA expanded this into a K-layer structure, represented as $(K \times \text{ResBlock} \rightarrow \text{Linear})$.

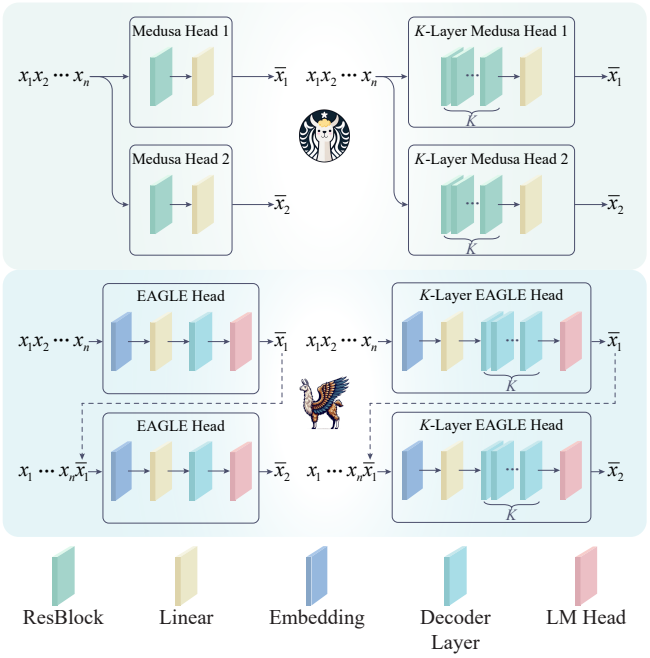

Fig. 2: Comparison of single-layer and multi-layer draft head structures. For each Medusa Head, KOALA expands the single ResBlock to K layers. In the EAGLE Head, KOALA extends the single Decoder Layer to K layers. For simplicity, each draft head predicts only the next two tokens, $\bar{x}_1$ and $\bar{x}_2$, based on the input sequence $x_1, x_2, \cdots, x_n$.

EAGLE Heads, in comparison, have a more complex structure. A conventional EAGLE Head consists of an Embedding, a Linear layer, a Decoder Layer, and an LM Head derived from the target LLM. The Embedding encodes historical tokens for autoregressive decoding, while the Linear layer integrates token and feature information before passing it to the Decoder Layer. The Decoder Layer then predicts features of subsequent tokens, which the LM Head maps to the vocabulary size. KOALA expanded this into a K-layer structure, represented as $(\text{Embedding} \rightarrow \text{Linear} \rightarrow K \times \text{Decoder Layer} \rightarrow \text{LM Head})$.

In summary, KOALA expands the single-layer draft head's prediction feature layer for subsequent tokens to K layers, while maintaining the structure of other data processing and mapping layers. Notably, for LLMs with more transformer layers, indicating a larger performance gap with single-layer draft heads, a higher K should be considered.

B. Training with Adversarial Learning

To improve the draft head's token prediction accuracy, we integrate a discriminator into the training process, combining adversarial learning with supervised training.

In adversarial learning, the generator and discriminator co-evolve, necessitating comparable capabilities. To align capabilities and optimize training outcomes, we select discriminators with layer counts matching those of the draft head. Furthermore, the primary objective of draft head training is to

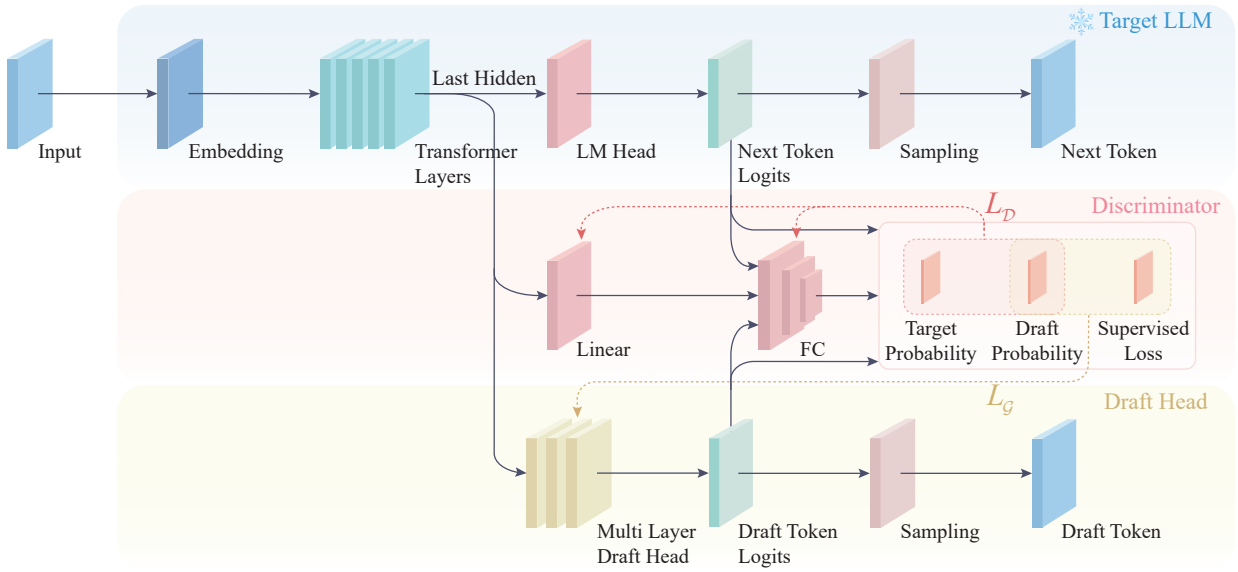

Fig. 3: Training process for multi-layer draft heads which incorporates adversarial learning into supervised training. The target LLM, featuring a snowflake logo, and its parameters remain unupdated throughout the process. The discriminator and draft head are trained adversarially, co-evolving until they reach a Nash equilibrium, whereupon the training terminates.

mirror the target LLM's functionality. To further unlock the draft head's potential, we implement distillation rather than using a fixed dataset for supervised training, a method proven effective for training draft models in speculative decoding [8].

Figure 3 illustrates the training process, comprising three main components: *Target LLM*, *Discriminator*, and *Draft Head*. The Target LLM provides input and real data for draft head training without parameter updates. The Draft Head ($\mathcal{G}$) takes the semantically rich final hidden states of the Target LLM as input. After autoregressive or non-autoregressive decoding through the multi-layer draft heads, whose parameters are the only ones updated in $\mathcal{G}$ throughout the training process, draft token logits are obtained, and the predicted token is generated through sampling. The Discriminator ($\mathcal{D}$) consists of a linear layer and a fully connected layer (FC). First, the linear layer processes the last hidden states from the Target LLM, mapping them to the same dimension as the token logits. Subsequently, based on the mapped last hidden states, the next token logits from the Target LLM, and the draft token logits from the Draft Head, the FC computes the *Target Probability* and *Draft Probability*, which represent the likelihoods that the input token logits originate from the Target LLM and Draft Head, respectively. In addition, $\mathcal{D}$ also calculates the *Supervised Loss* based on the next token logits and draft token logits, which serves as the supervised learning loss for distillation. Afterward, $\mathcal{D}$ updates its parameters based on the Target Probability and Draft Probability, while $\mathcal{G}$ updates its parameters using the Draft Probability and Supervised Loss. The loss functions $L_{\mathcal{G}}$ and $L_{\mathcal{D}}$ for $\mathcal{G}$ and $\mathcal{D}$ are presented in Equations 1 and 2, respectively.

$$L_{\mathcal{G}} = \underbrace{-\lambda\, \mathbb{E}_{\tilde{x}\sim p_d(d)}[\log(\mathcal{D}(\tilde{x}))]}_{\text{Adversarial Learning}} + \underbrace{L_{\text{Distill}}(d, q)}_{\text{Supervised Learning}} \quad (1)$$

$$L_{\mathcal{D}} = -\mathbb{E}_{\tilde{x}\sim p_d(d)}[\log(1-\mathcal{D}(\tilde{x}))] - \mathbb{E}_{\bar{x}\sim p_q(q)}\left[\log \mathcal{D}(\bar{x})\right] \quad (2)$$

Algorithm 1: Training Process for Draft Heads

Input: Multi-Layer Draft head $\mathcal{M}_d$, Target LLM output logits q, Input sequence $x_1, x_2, \cdots, x_n$

repeat
  ▷ **Draft Head Step**
  for *g-steps* **do**
    // $\mathcal{M}_d$ predicts logits for t subsequent tokens
    $d_1, d_2, \cdots, d_t \leftarrow \mathcal{M}_d(x \mid x_{\leq n})$;
    // Draft Head Back Forward Pass
    Compute $L_{\mathcal{G}} = \underbrace{-\lambda\, \mathbb{E}_{\tilde{x}\sim p_d(d_{\leq t})}[\log(\mathcal{D}(\tilde{x}))]}_{\text{Adversarial Learning}} + \underbrace{L_{\text{Distill}}(d_{\leq t}, q_{\leq t})}_{\text{Supervised Learning}}$;
    Update draft head parameters;
  end
  ▷ **Discriminator Step**
  for *d-steps* **do**
    // $\mathcal{M}_d$ predicts logits for t subsequent tokens
    $d_1, d_2, \cdots, d_t \leftarrow \mathcal{M}_d(x \mid x_{\leq n})$;
    // Discriminator Back Forward Pass
    Compute $L_{\mathcal{D}} = -\mathbb{E}_{\tilde{x}\sim p_d(d_{\leq t})}[\log(1-\mathcal{D}(\tilde{x}))] - \mathbb{E}_{\bar{x}\sim p_q(q_{\leq t})}\left[\log \mathcal{D}(\bar{x})\right]$;
    Update discriminator parameters;
  end
until *$\mathcal{G}$ and $\mathcal{D}$ reach a Nash equilibrium*;

Here, d and q represent the tokens logits predicted by the draft head and generated by the target LLM, respectively. λ denotes the weight of the adversarial learning loss function in $L_{\mathcal{G}}$. $L_{\text{Distill}}(\cdot)$ represents the supervised learning loss function in distillation, such as cross-entropy loss.

Once $\mathcal{G}$ and $\mathcal{D}$ reach a Nash equilibrium, the training is deemed complete. Algorithm 1 summarizes the entire training process.

III. Experiments

A. Experimental Setup

To assess KOALA's efficiency, we utilize Medusa and EAGLE as representatives of non-autoregressive and autoregressive draft heads, respectively, with Vicuna models (7B, 13B, 33B) [9] serving as target LLMs. Training utilizes the ShareGPT dataset with 68,000 dialogue iterations. Evaluations are performed on an A800 80G GPU using MT-Bench [10], a multi-turn conversation benchmark encompassing diverse tasks such as mathematical analysis, abstract extraction, and code generation. All experiments employ a greedy decoding strategy, accepting tokens only when they match the target LLM's greedy next-token generation.

Medusa and EAGLE layers are configured with K = 1, 2, 3, while the discriminator's FC layers range from 1 to 3, with learning rates between [1e-5, 5e-4]. The adversarial learning loss function weight λ in Equation 1 is set within the range [0.05, 0.5]. Both the draft head and discriminator are set to perform one iteration ($g = d = 1$). The evaluation is conducted with a batch size of 1. For fair comparison, the original Medusa and EAGLE are trained using knowledge distillation. All other parameters and training settings adhere to the original Medusa and EAGLE configurations.

The following metrics are employed to evaluate KOALA:

- Walltime speedup ratio: The speedup ratio achieved by the draft head compared to vanilla autoregressive decoding, serving as the primary performance metric.
- Average acceptance length ℓ: The average number of tokens generated per forward pass by the target LLM equipped with the draft head. Higher ℓ values indicate improved draft head prediction accuracy.
- Acceptance rate n-α: The draft head's accuracy in predicting the nth subsequent token. Following the original EAGLE settings, we use chain drafts without tree attention, evaluating the prediction accuracy for the first three tokens (n = 1, 2, 3).

B. Main Results

Figure 4 and Table I demonstrate the effectiveness of KOALA. We iterated through draft heads' layers K from 1 to 3 and reported the highest speedup ratio. Compared with Medusa and EAGLE, representatives of non-autoregressive and autoregressive draft heads respectively, KOALA optimization improves the speedup ratio by 0.24x-0.29x and 0.35x-0.41x, which are 10.57%-12.83% and 11.55%-14.09% faster than their original draft heads. These results validate KOALA's efficacy for both non-autoregressive and autoregressive draft heads. The enhanced performance stems from the target LLM's increased acceptance rate of tokens predicted by the draft head. Specifically, the number of tokens generated per forward pass rises by 0.26-0.45, resulting in fewer iterations in the speculative decoding algorithm and consequently faster LLM inference.

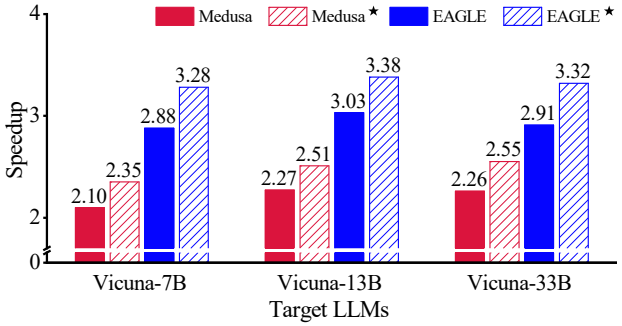

Fig. 4: Speedup ratios of Medusa, EAGLE, and their KOALA-optimized versions achieving maximum speedup improvement, denoted by superscript ★. All configurations achieve maximum speedup at K = 2, except Medusa on Vicuna-33B, which peaks at K = 3.

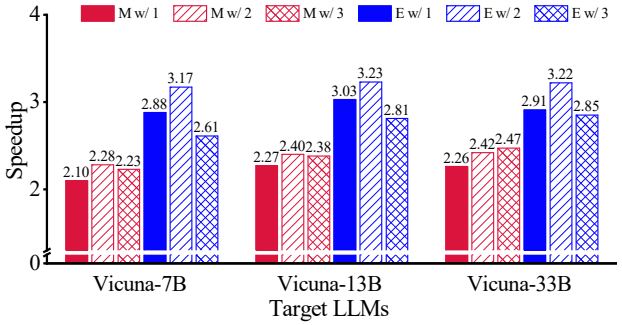

Fig. 5: Speedup ratios of Medusa and EAGLE with varying layer structures. "M w/ 1" and "E w/ 1" represent the original single-layer Medusa and EAGLE, respectively.

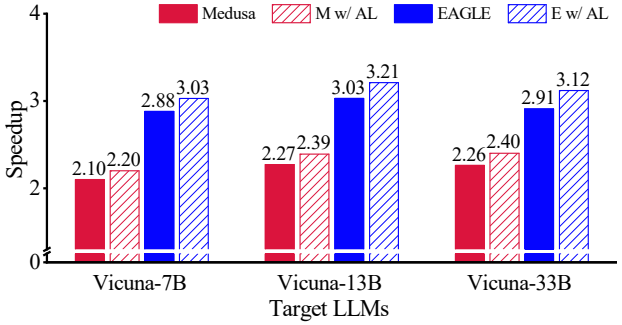

Fig. 6: Speedup ratios of Medusa, EAGLE, and their variants incorporating adversarial learning during training.

C. Ablation Study

1) Multi-Layer: KOALA transformers the traditional single-layer draft head into a multi-layer architecture. Figure 5 and Table I illustrate the performance comparison between multi-layer architecture (K = 2, 3) and the original single-layer architecture (K = 1), demonstrating the impact of using multi-layer approach. Compared with the original single-layer Medusa and EAGLE, the multi-layer architecture increases the average acceptance length by 0.18-0.45 and the speedup ratio by 0.11x-0.31x, indicating that the multi-layer architecture enables the draft head to better mirror the functionality of the target LLM. Notably, while the token acceptance rate and average acceptance length increase with K, the optimal

TABLE I: Average acceptance lengths ℓ and acceptance rates n-α of Medusa, EAGLE, and their variants on Vicuna models. "V" represents Vicuna. "M" and "E" denote Medusa and EAGLE, respectively. "w/ AL" indicates the draft head incorporating adversarial learning during training. "w/ 2" and "w/ 3" signify draft heads using 2-layer and 3-layer architectures, respectively. The superscript ★ indicates the KOALA-optimized draft heads yielding the maximum speedup improvement in Figure 4.

	Model	Medusa	M w/ AL	M w/ 2	M w/ 3	Medusa★	EAGLE	E w/ AL	E w/ 2	E w/ 3	EAGLE★
ℓ	V 7B	2.62	2.70	2.82	2.87	**2.88**	3.91	4.00	4.20	**4.36**	4.28
	V 13B	2.69	2.74	2.87	2.94	**2.95**	3.96	4.04	4.24	**4.38**	4.33
	V 33B	2.52	2.58	2.70	2.90	**2.97**	3.78	3.84	4.10	**4.20**	4.16
1-α	V 7B	0.57	0.58	0.59	0.60	**0.60**	0.79	0.79	0.81	0.82	**0.82**
	V 13B	0.58	0.59	0.61	0.62	**0.63**	0.79	0.79	0.81	0.83	**0.83**
	V 33B	0.55	0.57	0.59	0.61	**0.64**	0.77	0.78	0.80	**0.81**	0.81
2-α	V 7B	0.40	0.41	0.41	0.43	**0.43**	0.74	0.75	0.77	**0.79**	0.77
	V 13B	0.41	0.41	0.43	**0.44**	0.43	0.74	0.75	0.79	**0.80**	0.79
	V 33B	0.40	0.40	0.40	**0.42**	0.42	0.72	0.74	0.73	**0.76**	0.75
3-α	V 7B	0.31	0.31	0.32	0.32	**0.34**	0.69	0.71	0.74	0.75	**0.75**
	V 13B	0.31	0.33	0.33	0.34	**0.34**	0.70	0.71	0.75	**0.76**	0.75
	V 33B	0.30	0.30	0.31	**0.32**	0.32	0.68	0.67	0.69	0.71	**0.71**

speedup for most Medusa or EAGLE is achieved at $K = 2$, with the exception of Medusa at $K = 3$ on Vicuna 33B. This phenomenon is attributed to the increased number of draft head parameters in the multi-layer structure, which introduces additional drafting overhead. Consequently, it is crucial to balance the improved prediction accuracy against the increased drafting overhead by selecting an appropriate K. For Medusa and EAGLE, the multi-layer architecture achieves the most significant speedup improvements on the Vicuna 33B model, reaching 0.21x and 0.31x, respectively. This is attributed to the multi-layer architecture enhancing draft head performance by narrowing the parameter-induced performance gap between the draft head and the target LLM. Furthermore, in this experiment, the 33B model, containing the most transformer layers, exhibits the most pronounced performance disparity compared to the original single-layer draft head. Additionally, the speedup ratio of the draft head with $K = 3$ improves as the target LLM size increases. Specifically, for Medusa, the speedup with $K = 3$ shifts from near-optimal to optimal when moving from Vicuna 7B to Vicuna 33B. For EAGLE, although $K = 3$ has not yet reached optimal performance, the gap is narrowing. We speculate that as the target LLM size further increases, EAGLE with $K = 3$ or higher will yield optimal results. Consequently, higher K values should be considered for larger target LLMs.

2) Adversarial Learning: Another innovation of KOALA is the incorporation of adversarial learning into the conventional supervised training process for draft heads. Figures 6 and Table I illustrate the comparative results, showcasing the impact of the adversarial learning approach. Compared to the original Medusa and EAGLE, the integration of adversarial learning increases the average acceptance length by 0.06-0.1 and improves the speedup ratio by 0.1x-0.19x. These enhancements indicate that adversarial learning effectively improves the prediction accuracy of draft heads, thereby enhancing speculative decoding. Notably, unlike the multi-layer structure, adversarial learning does not alter the original draft head architecture, thereby incurring no additional drafting overhead. Consequently, any enhancement in the draft head's prediction accuracy directly contributes to improved speedup performance. Interestingly, we observe that EAGLE demonstrates more substantial improvements compared to Medusa. This discrepancy may be attributed to the limited number of training epochs in Medusa's original configuration, potentially impeding the draft head and discriminator from reaching Nash equilibrium. Conversely, EAGLE's longer training period enables it to more fully exploit the potential of adversarial learning.

IV. Related Work

Recent research has explored diverse approaches to enhance LLM inference efficiency, including quantization [11], network pruning [12], and attention simplification [13]. These methods accelerate processing by either reducing computational precision or minimizing operational complexity. Furthermore, architectural optimization strategies for LLM inference have emerged, including non-autoregressive decoding [14], early exiting [15], and knowledge distillation [9]. While these approaches substantially accelerate LLM inference, they often compromise generation quality.

Speculative decoding can achieve lossless acceleration through the draft-then-verify paradigm. Blockwise Decoding [14] pioneered the integration of additional feedforward networks (FFNs) into transformer decoders, accelerating greedy decoding through enhanced parallel generation. Speculative Sampling [4], [5] extends this concept beyond greedy decoding to non-greedy methods, while preserving the output distribution of the original sampling approach.

Existing speculative decoding strategies can be classified into two main categories: independent drafting and self-drafting techniques. SpecDec [16] employs a non-autoregressive independent drafter that delivers significant speedup but demands substantial training overhead. A more cost-effective approach leverages smaller LMs to accelerate larger models within the same model series [17]. However, cross-series model coordination presents significant challenges due to architectural and implementation disparities.

Self-drafting addresses these challenges by utilizing the target LLM itself for prediction, eliminating the need for additional draft models. One approach incorporates an early exit mechanism during decoding for advance token prediction [18]. Another strategy enables adaptive layer skipping during inference to optimize computational efficiency [19]. Recent advances integrate lightweight non-autoregressive or autoregressive prediction heads after the target LLM's final hidden states to leverage rich semantic information for next-token prediction. Medusa [6] introduces multiple non-autoregressive draft heads after the final hidden states to generate candidate tokens in parallel, further exploiting the potential of FFNs and advancing non-autoregressive methods. Amphista [20] enhances Medusa by introducing an automatic embedding block with a bidirectional self-attention module and a staged adaptation layer for feature transformation. Hydra [21] leverages previously predicted token information to transform non-autoregressive draft heads into an autoregressive FFN. Clover [22] enhances the prediction accuracy of regressive draft heads by incorporating sequential knowledge through regression connections, attention decoders, and enhancement modules. EAGLE [7] integrates token and feature information to transform the FFN into an autoregressive head, which consists of a fully connected layer and a decoder layer, thereby significantly improving the acceptance rate of draft tokens. EAGLE-2 [23] dynamically adjusts the draft tree structure based on the confidence score of the draft model, further enhancing the inference efficiency of LLMs. In contrast, KOALA transforms the traditional single-layer draft head into a multi-layer structure and incorporates adversarial learning into conventional supervised training. This approach enables the draft head to more closely mirror the functionality of the target LLM, thereby enhancing speculative decoding.

V. Conclusion

In this paper, we introduce KOALA, an efficient orthogonal approach for draft head optimization that enhances speculative decoding for LLMs. KOALA transforms the traditional single-layer draft head into a multi-layer structure and incorporates adversarial learning into conventional supervised training. At the cost of a slight increase in drafting overhead, KOALA enables the draft head to more closely mirror the functionality of LLMs, thereby accelerating LLM inference. We conducted comprehensive evaluations of KOALA on Medusa and EAGLE, representing non-autoregressive and autoregressive draft heads, respectively, using Vicuna models (7B, 13B, 33B) as target LLMs and MT-bench dataset for assessment. KOALA achieves a 0.24x-0.41x improvement in latency speedup ratio, which is 10.57%-14.09% faster than the original draft heads.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[3] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen *et al.*, "Palm 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[4] Y. Leviathan, M. Kalman, and Y. Matias, "Fast inference from transformers via speculative decoding," in *International Conference on Machine Learning*. PMLR, 2023, pp. 19 274–19 286.

[5] C. Chen, S. Borgeaud, G. Irving, J.-B. Lespiau, L. Sifre, and J. Jumper, "Accelerating large language model decoding with speculative sampling," *arXiv preprint arXiv:2302.01318*, 2023.

[6] T. Cai, Y. Li, Z. Geng, H. Peng, J. D. Lee, D. Chen, and T. Dao, "Medusa: Simple LLM inference acceleration framework with multiple decoding heads," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=PEpbUobfJv

[7] Y. Li, F. Wei, C. Zhang, and H. Zhang, "EAGLE: Speculative sampling requires rethinking feature uncertainty," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=1NdN7eXyb4

[8] Y. Zhou, K. Lyu, A. S. Rawat, A. K. Menon, A. Rostamizadeh, S. Kumar, J.-F. Kagy, and R. Agarwal, "Distillspec: Improving speculative decoding via knowledge distillation," *arXiv preprint arXiv:2310.08461*, 2023.

[9] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, vol. 2, no. 3, p. 6, 2023.

[10] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[11] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[12] E. Frantar and D. Alistarh, "Sparsegpt: Massive language models can be accurately pruned in one-shot," in *International Conference on Machine Learning*. PMLR, 2023, pp. 10 323–10 337.

[13] Z. Zhang, Y. Sheng, T. Zhou, T. Chen, L. Zheng, R. Cai, Z. Song, Y. Tian, C. Ré, C. Barrett *et al.*, "H2o: Heavy-hitter oracle for efficient generative inference of large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[14] M. Stern, N. Shazeer, and J. Uszkoreit, "Blockwise parallel decoding for deep autoregressive models," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[15] W. Zhou, C. Xu, T. Ge, J. McAuley, K. Xu, and F. Wei, "Bert loses patience: Fast and robust inference with early exit," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 330–18 341, 2020.

[16] H. Xia, T. Ge, P. Wang, S.-Q. Chen, F. Wei, and Z. Sui, "Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 3909–3925.

[17] Z. Sun, A. T. Suresh, J. H. Ro, A. Beirami, H. Jain, and F. Yu, "Spectr: Fast speculative decoding via optimal transport," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[18] S. Yang, G. Lee, J. Cho, D. Papailiopoulos, and K. Lee, "Predictive pipelined decoding: A compute-latency trade-off for exact llm decoding," *arXiv preprint arXiv:2307.05908*, 2023.

[19] J. Zhang, J. Wang, H. Li, L. Shou, K. Chen, G. Chen, and S. Mehrotra, "Draft & verify: Lossless large language model acceleration via self-speculative decoding," *arXiv preprint arXiv:2309.08168*, 2023.

[20] Z. Li, X. Yang, Z. Gao, J. Liu, Z. Liu, D. Li, J. Peng, L. Tian, and E. Barsoum, "Amphista: Accelerate llm inference with bi-directional multiple drafting heads in a non-autoregressive style," *arXiv preprint arXiv:2406.13170*, 2024.

[21] Z. Ankner, R. Parthasarathy, A. Nrusimha, C. Rinard, J. Ragan-Kelley, and W. Brandon, "Hydra: Sequentially-dependent draft heads for medusa decoding," *arXiv preprint arXiv:2402.05109*, 2024.

[22] B. Xiao, C. Shi, X. Nie, F. Yang, X. Deng, L. Su, W. Chen, and B. Cui, "Clover: Regressive lightweight speculative decoding with sequential knowledge," *arXiv preprint arXiv:2405.00263*, 2024.

[23] Y. Li, F. Wei, C. Zhang, and H. Zhang, "Eagle-2: Faster inference of language models with dynamic draft trees," *arXiv preprint arXiv:2406.16858*, 2024.