Halfway Speculative Decoding: Direct Acceptance Rate Optimization with Joint Drafter-Target Training

Marian-Sergiu Nistor
Independent Researcher
Iași, Romania
sergiunistor.info@gmail.com

Abstract

In speculative decoding, direct optimization of the acceptance rate through LK losses (Samarin et al., 2026) has proven to be more effective than the standard approach of training the draft head through KL divergence minimization against the target's output distribution, as the acceptance rate is what ultimately governs inference speedup. However, prior approaches have either jointly trained the draft head and target model without directly targeting acceptance rate, or optimized the acceptance rate on the draft head alone while keeping the target frozen. We propose **Halfway Speculative Decoding**, which extends LK loss optimization to both models jointly, using cross-entropy loss to regularize the target model and prevent quality degradation. We study how speculation metrics and target quality scale with training budget by training separate model instances on increasing numbers of training examples and evaluating across several benchmarks.

1 Introduction

The speculative decoding acceptance rate is governed by the TV distance between the draft and target distributions. However, the standard training objective has been using KL divergence as a proxy. While minimizing KL divergence does reduce distributional discrepancy, draft heads converge to suboptimal points where KL progress does not guarantee TV improvement.

Furthermore, most prior approaches shift only the drafter distribution. Some train the draft head conditioned on the target's hidden states, jointly optimizing a feature regression and next-token prediction loss (Li et al., 2024b, 2025b). Similarly, others align the drafter with the target via knowledge distillation (Zhou et al., 2024). More recent methods directly optimize the acceptance rate as the training objective (Samarin et al., 2026). Although some approaches jointly train both models (Cai et al., 2024), none of them directly targets the acceptance rate on both the drafter and the target simultaneously.

Method	Loss	MT-bench τ	HumanEval τ	GSM8K τ
EAGLE-3 (Temp = 0)	KL	3.75	4.82	4.50
	TV	2.81	3.42	3.34
	$\mathcal{L}_{\text{LK}}^{\alpha}$	3.77	4.82	4.55
	$\mathcal{L}_{\text{LK}}^{\lambda}, \lambda = 0.5$	3.78	4.84	4.53
	$\mathcal{L}_{\text{LK}}^{\lambda}, \eta = 1$	3.80	4.83	4.54
	$\mathcal{L}_{\text{LK}}^{\lambda}, \eta = 3$	3.84	**4.89**	**4.57**
MEDUSA (Temp = 0)	KL	2.05	2.41	2.11
	$\mathcal{L}_{\text{LK}}^{\alpha}$	2.06	2.42	2.11
	$\mathcal{L}_{\text{LK}}^{\lambda}, \eta = 10$	2.07	2.44	2.13
MLP (Temp = 0)	KL	2.45	2.42	2.42
	$\mathcal{L}_{\text{LK}}^{\alpha}$	2.48	2.46	2.46
	$\mathcal{L}_{\text{LK}}^{\lambda}, \eta = 3$	2.48	2.83	2.46
EAGLE-3 (Temp = 1)	KL	3.39	4.31	3.88
	TV	2.67	3.25	3.12
	$\mathcal{L}_{\text{LK}}^{\alpha}$	3.50	4.48	3.98
	$\mathcal{L}_{\text{LK}}^{\lambda}, \lambda = 0.5$	3.35	4.36	3.95
	$\mathcal{L}_{\text{LK}}^{\lambda}, \eta = 1$	3.51	4.47	3.96
	$\mathcal{L}_{\text{LK}}^{\lambda}, \eta = 3$	3.48	4.52	4.02
MEDUSA (Temp = 1)	KL	1.72	2.02	1.81
	$\mathcal{L}_{\text{LK}}^{\alpha}$	1.78	2.09	1.85
	$\mathcal{L}_{\text{LK}}^{\lambda}, \eta = 10$	1.85	2.22	1.92
MLP (Temp = 1)	KL	2.13	2.16	2.16
	$\mathcal{L}_{\text{LK}}^{\alpha}$	2.17	2.19	2.19
	$\mathcal{L}_{\text{LK}}^{\lambda}, \eta = 3$	2.19	2.62	2.18
Ours	$\mathcal{L}_{\text{LK}}^{\lambda}, \eta = 3$	**3.88**	4.01	3.65

Table 1: Comparison of τ across methods, losses, benchmarks, and temperatures. Benchmarks span MT-bench (Zheng et al., 2023), HumanEval (Chen et al., 2021), and GSM8K (Cobbe et al., 2021). We evaluate our method (using 60k training examples) alongside EAGLE-3 (Li et al., 2025b), MEDUSA (Cai et al., 2024), and the multi-stage MLP architecture introduced by Wertheimer et al. (2024), with baseline results sourced from Samarin et al. (2026). All methods, including ours, use Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as the target model. Our method achieves the best results on MT-bench, while remaining competitive on HumanEval and GSM8K.

Shifting the target distribution toward the drafter is a natural extension, but pulling a stronger model toward a weaker one risks degrading its generation quality. We address this using three mechanisms, in order to prevent catastrophic forgetting in the target model:

- We regularize the target using cross-entropy loss during training.
- We limit the training budget to 60k examples.
- We train on data generated by the target model itself, following the approach used by Samarin et al. (2026).

Thus, we propose **Halfway Speculative Decoding**, which trains an EAGLE-3-style draft head (Li et al., 2025b) and fine-tunes the target model via LoRA (Hu et al., 2022), jointly optimizing both on the LK loss (Samarin et al., 2026), while regularizing the target with a weighted cross-entropy loss. We use Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as the target model, trained on the Llama-3.1-8B-Instruct-Infinity-Instruct-0625 dataset (Samarin et al., 2026), a collection of prompt-response pairs sampled from Infinity-Instruct (Li et al., 2025a) using the target model itself.

We show that our method improves the average acceptance length while preserving the target quality, with gains that hold across MT-bench (Zheng et al., 2023), HumanEval (Chen et al., 2021), GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2024), GPQA (Rein et al., 2024), MMLU-Pro (Wang et al., 2024), and BigCodeBench (Zhuo et al., 2025). We further analyze how the acceptance metrics and target performance scale with training budget by evaluating model checkpoints at increasing numbers of training steps.

2 Related Work

In this section, we discuss prior work in the speculative decoding space, covering drafter-only training, joint drafter-target training, direct acceptance rate optimization, and alternative speculative decoding methods.

Drafter-Only Training

EAGLE (Li et al., 2024b) and EAGLE-3 (Li et al., 2025b) train the draft head conditioned on the target's hidden states, jointly minimizing a feature regression loss and a next-token cross-entropy loss against the target's predictions, while keeping the target frozen. DistillSpec (Zhou et al., 2024) similarly applies knowledge distillation to align the drafter with the target distribution. Both lines of work shift only the draft distribution, leaving the target unchanged. Our approach differs by also training the target model, pulling both distributions toward each other.

Joint Drafter-Target Training

MEDUSA-2 (Cai et al., 2024) jointly trains the target model alongside its draft heads, showing that updating the target improves the accuracy of the draft model. Both models minimize cross-entropy on the same dataset, using distributional alignment as a surrogate for acceptance rate. While this encourages both distributions to converge toward the same targets, it does not directly optimize the speculation quality, since there is no guarantee the two distributions converge toward the same optimum. We build on this idea by replacing the training objective with the LK loss, directly targeting acceptance rate on both models.

Direct Acceptance Rate Optimization

LK loss (Samarin et al., 2026) directly optimizes the acceptance rate rather than using KL divergence as a proxy, consistently outperforming KL-based training across architectures and model scales. However, it only trains the draft head while keeping the target frozen. We extend this to joint training, applying LK loss to both the draft head and the target model, with cross-entropy regularization on the target to preserve its generation quality and prevent catastrophic forgetting.

Beyond Distribution-Aligned Verification

Judge Decoding (Bachmann et al., 2025) tackles the same root problem of the target rejecting too many valid draft tokens, but from a different angle. Instead of modifying the draft or target models themselves, it trains a lightweight linear head on top of the target's frozen embeddings to predict token correctness, replacing logit-based verification. The tradeoff is that it violates the lossless output distribution guarantee, and in the worst case the judge accepts factually incorrect tokens that the target model would have caught.

3 Method

We present our joint training approach for directly optimizing acceptance rate on both the draft head and the target model simultaneously, along with

the mechanisms we use to prevent target model degradation.

3.1 The LK Loss

We leverage the LK loss introduced by Samarin et al. (2026), which directly targets the acceptance rate rather than using KL divergence as a proxy. Let p denote the target distribution and q the draft distribution. We use the hybrid variant $\mathcal{L}_{\text{LK}}^{\lambda}$, which combines KL divergence with TV distance under an adaptive schedule:

$$\begin{aligned}\mathcal{L}_{\text{LK}}^{\lambda}(p,q) = \lambda(\alpha)\cdot \text{KL}(p\|q) \\ + (1-\lambda(\alpha))\cdot \text{TV}(p,q)\end{aligned} \tag{1}$$

where α is the per-token acceptance rate, defined as the expected token-level overlap between draft and target distributions:

$$\alpha = \sum_{v\in\mathcal{V}} \min(q(v), p(v)) \tag{2}$$

and $\lambda(\alpha)$ is an adaptive weight that controls the balance between KL and TV:

$$\lambda(\alpha) = \exp\big(-\eta\cdot \text{sg}[\alpha]\big), \quad \eta > 0 \tag{3}$$

Here $\text{sg}[\cdot]$ is a stop-gradient operator applied to prevent gradients from flowing through λ during backpropagation, and η controls the rate of transition. Early in training, when α is low, λ converges to 1 and the objective reduces to KL minimization, which provides well-conditioned gradients from a randomly initialized draft. As training progresses and α increases, λ decays toward zero, shifting the focus to TV minimization, which directly targets acceptance rate. This curriculum behavior is what distinguishes $\mathcal{L}_{\text{LK}}^{\lambda}$ from a fixed mixture of objectives.

For multi-step drafting with K draft positions, the loss aggregates across positions with an exponential decay that prioritizes earlier positions, which have the largest impact on average acceptance length:

$$\hat{\mathcal{L}}_{\text{LK}}^{\lambda} = \sum_{k=1}^{K} \gamma^{k-1}\cdot \mathcal{L}_{\text{LK}}^{\lambda}(p, q^{(k)}) \tag{4}$$

where $q^{(k)}$ is the draft distribution at position k, and $\gamma \in (0, 1]$ is the per-step discount factor.

3.2 Joint Training Objective

Prior work applies the LK loss only to the draft head while keeping the target frozen. We instead train both the draft head and the target model on the LK loss, directly optimizing for acceptance rate.

However, as previously stated, pulling a stronger model toward a weaker one degrades its generation quality. We address this by adding a cross-entropy regularization term that anchors p to the training data distribution, preventing it from collapsing toward q. This term has no gradient path into the draft head, acting solely on the target. The full target training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{LK}}^{\lambda} + \lambda_{\text{CE}}\cdot \mathcal{L}_{\text{CE}} \tag{5}$$

where $\lambda_{\text{CE}} \geq 0$ controls the regularization strength and $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss over tokens computed from the target model's output logits. $\mathcal{L}_{\text{LK}}^{\lambda}$ backpropagates through both the draft head and the target model's trainable parameters, while $\mathcal{L}_{\text{CE}}$ backpropagates only through the target.

3.3 Draft Head

We use an EAGLE-3-style draft head (Li et al., 2025b) trained from scratch jointly with the target model. At each forward pass, the head constructs an initial hidden state by concatenating the target model's embedding layer output with hidden states from intermediate target layers 4, 16, and 32, and projecting the result via a learned linear layer. It then autoregressively generates K draft tokens by repeatedly applying a single transformer layer: at each step, the transformer layer processes the current hidden state to produce draft logits, and the hidden state for the next step is computed by fusing the current output with a time-shifted version of the input token embeddings via a second learned linear layer.

3.4 Preventing Target Model Degradation

Directly fine-tuning all target model parameters would be prohibitively expensive and would risk rapid forgetting of pretrained knowledge. We instead adapt the target model using LoRA adapters (Hu et al., 2022), which introduce trainable low-rank matrices alongside the frozen weights of the attention and feed-forward projection matrices, keeping the base model weights unchanged. This substantially reduces the number of trainable parameters on the target side, making joint training computationally feasible and limiting the surface area over which drift can occur.

Additionally, training data construction follows the approach described in Samarin et al. (2026), generating responses using the target model itself. We leverage this property to anchor the target model during joint training, using its own outputs as a reference that reduces the risk of drifting under the joint objective.

Beyond parameter efficiency, we also constrain the training duration. Even with cross-entropy regularization, prolonged training on the joint objective risks gradually eroding the target model's pretrained knowledge. We study the effect of training budget explicitly by training separate model instances on increasing numbers of examples and evaluating each, rather than checkpointing a single run. This gives a cleaner signal of how acceptance metrics and target quality scale independently with training compute, without conflating within-run dynamics across different stages.

4 Experimental Setup

In the following section, we describe the models, training data, and evaluation protocol used throughout our experiments. All experiments are carried out with a single target model and draft head pair, with the joint training objective described in Section 3.

4.1 Models

We use Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as the target model throughout all experiments, adapted via LoRA (Hu et al., 2022) applied to all attention and feed-forward projection matrices with rank $r = 16$, $\alpha = 32$, and dropout 0.05. The draft head follows an EAGLE-3-style architecture (Li et al., 2025b), initialized randomly and trained from scratch together with the target model.

4.2 Training Dataset

We train on the Llama-3.1-8B-Instruct-Infinity-Instruct-0625 dataset (Samarin et al., 2026), a collection of 660K prompt-response pairs constructed by generating responses to Infinity-Instruct prompts (Li et al., 2025a) using the target model itself at temperature $T = 1$. Training on target-generated responses serves as an anchor for the target model, reducing the risk of drifting away from its original output distribution. We use a maximum sequence length of 512 tokens and mask prompt tokens in all loss computations, propagating gradients only over model response tokens.

4.3 Training Configuration

We train with a batch size of 64, a learning rate of 4×10^{-4} with cosine scheduling and 100 warmup steps, using the AdamW optimizer with $(\beta_1, \beta_2) = (0.9, 0.95)$, and gradient clipping at 0.5. The same learning rate is used for both the target LoRA and the draft head. We set $K = 7$ draft positions and LK loss hyperparameters $\eta = 3$, $\gamma = 0.8$, $\lambda_{CE} = 1$, and $\varepsilon = 10^{-6}$, following Samarin et al. (2026). All experiments use *bfloat16* mixed precision.

4.4 Evaluation Benchmarks

We evaluate on seven benchmarks spanning diverse domains: MT-bench (Zheng et al., 2023), HumanEval (Chen et al., 2021), GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2024), GPQA Diamond (Rein et al., 2024), MMLU-Pro (Wang et al., 2024), and BigCodeBench (Zhuo et al., 2025). We report the average acceptance length (τ) on all seven benchmarks, following the evaluation protocol of Samarin et al. (2026) with chain sampling over $K = 7$ steps at temperature $T = 1$.

For target model quality, we evaluate on all benchmarks except MT-bench, reporting perplexity. We also report accuracy on multiple-choice benchmarks (GPQA Diamond, MMLU-Pro) by selecting the candidate answer with the lowest negative log-likelihood under the target model.

4.5 Evaluating Training Dynamics

We evaluate two quantities throughout training: the average acceptance length of the joint system and the degree of drift in the target model's output distribution. To study how each metric scales with training data, we train separate model instances on 10K, 20K, 30K, 40K, 50K, and 60K training examples, rather than evaluating checkpoints from a single run, giving a cleaner signal unconfounded by within-run learning dynamics. Each instance is trained from scratch with the same configuration and evaluated independently, allowing us to directly observe whether gains in acceptance rate come at the cost of target quality degradation.

Acceptance Rate

We report the expected number of tokens accepted per speculation round, τ, following the convention of Leviathan et al. (2023) and Samarin et al. (2026):

$$\tau = K \cdot \frac{\text{\# accepted tokens}}{\text{\# drafted tokens}} + 1 \qquad (6)$$

The $+1$ accounts for the bonus token that is always sampled from the target distribution after each verification round, guaranteeing at least one new token per round regardless of draft quality. We evaluate using chain sampling at temperature $T = 1$.

Target Model Drift

To measure whether joint training degrades the target model, we evaluate its output distribution independently of the draft head, comparing the original model against the LoRA-adapted version after joint training. We report perplexity and, where applicable, accuracy on the benchmark examples, with prompt tokens masked, as a measure of how much the target's predicted distribution has shifted from its original state.

5 Results

We present results across three axes:

- We compare mean accepted tokens per speculation round (τ) against prior methods.
- We analyze how τ and target model drift scale with training budget.
- We study the evolution of the training dynamics throughout a training run.

All scaling experiments use separate model instances trained to each data scale rather than checkpoints of a single run, with the configuration described in Section 4.

Empirical Comparison

Table 1 compares τ across several methods and benchmarks. Our method achieves the highest τ value on MT-bench (Zheng et al., 2023) and competitive results across HumanEval (Chen et al., 2021) and GSM8K (Cobbe et al., 2021).

Notably, this result is achieved using only 60K training examples, approximately 9% of the 660K examples used by Samarin et al. (2026), suggesting that jointly optimizing acceptance rate on both the draft head and the target model is a significantly more data-efficient approach than drafter-only training. We expect that scaling to the full training set would yield further improvements.

Draft Token Acceptance Evolution

Figure 1 shows how τ evolves as a function of training examples across all benchmarks. Mean accepted tokens per speculation round increase consistently with training data across all evaluated

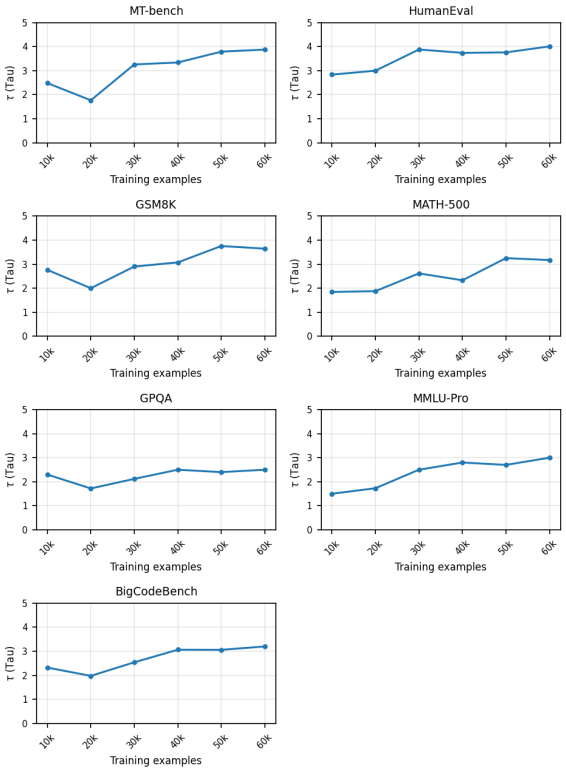

Figure 1: τ as a function of the number of training examples seen during fine-tuning, evaluated using separate model instances trained for each data scale, across multiple benchmarks. τ consistently increases as more training examples are seen.

benchmarks, with no signs of saturation at 60K examples.

This suggests further gains may be achievable with more training data, and that the joint objective does not cause τ to plateau prematurely as a result of competing gradients from the cross-entropy regularization term.

Target Model Drift

Figure 2 shows perplexity across benchmarks at increasing data scales, with higher variance early in training that settles as more examples are seen. This suggests the target model's output distribution undergoes some initial adjustment before stabilizing under the joint objective. Together, these results indicate that τ can be improved substantially without meaningfully compromising the target model's generation quality at sufficient training scale.

Figure 3 highlights the accuracy across benchmarks as a function of the training data scale. A slight downward trend is visible as more training examples are being used, consistent with the ex-

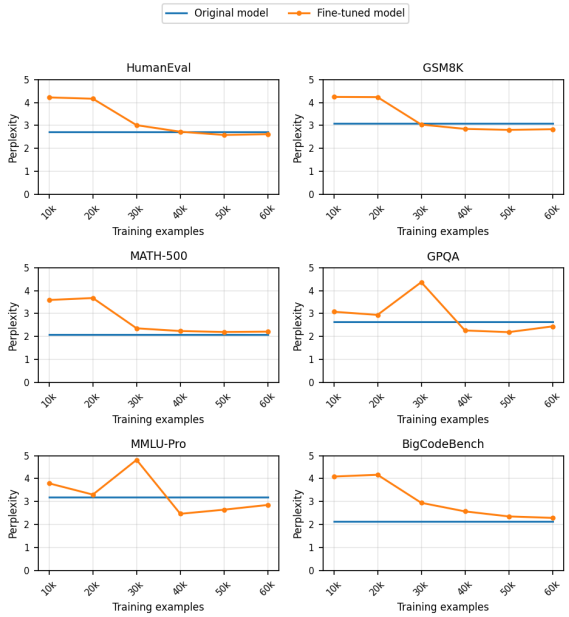

Figure 2: Perplexity trends across benchmarks as each independently trained model sees a progressively larger portion of the training set. Perplexity remains roughly stable at larger data scales.

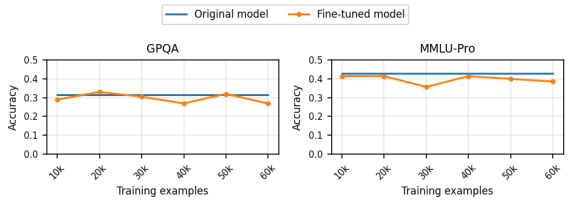

Figure 3: Benchmark accuracy across the evaluated tasks, measured after fine-tuning independent model instances on progressively larger training subsets. Accuracy shows a slight downward trend as more training examples are seen.

pected effect of the joint objective pulling the target distribution toward the weaker draft distribution. However, the degradation is modest across all benchmarks, indicating that the cross-entropy regularization and limited training budget are effective at constraining drift, anchoring the target distribution.

Training Dynamics

Figure 4 shows the evolution of the training losses over the 60K-example run. The fact that both the LK loss and the cross-entropy loss decrease in unison indicates that the two objectives do not conflict during training, and that improving acceptance rate does not come at the cost of increased cross-entropy on the training data.

The flattening behavior observed toward the

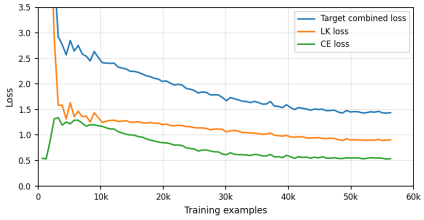

Figure 4: Evolution of training losses during the 60K-example run, showing that both individual losses decrease simultaneously without conflicting.

end of training suggests that the current configuration approaches saturation around 55K examples. Whether training beyond this point yields further gains likely depends on a range of factors including the regularization strength, the LK loss hyperparameters, the choice of training objective, the draft head architecture, and the target model itself. A systematic ablation over these dimensions is left for future work.

6 Conclusion

We proposed **Halfway Speculative Decoding**, a joint training framework that directly optimizes the acceptance rate on both the draft head and the target model simultaneously, extending the LK loss (Samarin et al., 2026) to both sides of the speculative decoding system. We adapt the target model via LoRA to keep joint training computationally feasible, and prevent target model degradation through cross-entropy regularization, a constrained training budget, and training on target-generated responses to anchor the target's output distribution.

Our results show that jointly optimizing acceptance rate on both models is substantially more data-efficient than drafter-only training: using only 60K training examples, approximately 9% of the data used by prior work, our method achieves competitive or higher τ across the evaluated benchmarks. Target model drift remains modest throughout training, with perplexity staying stable at a sufficient training data scale, indicating that acceptance rate gains do not come at the cost of generation quality.

Several directions remain for future work. Extending joint training to tree-based drafting methods (Miao et al., 2024; Cai et al., 2024; Li et al., 2024a) is a natural next step, as the current evaluation uses chain sampling only. Also, evaluating across different target model families would clarify how broadly the joint training benefits generalize.

Limitations

All experiments are conducted with a single target model and draft head pair, and it remains unclear whether the observed gains transfer to other model families. We also evaluate our method using chain sampling only, whereas some speculative decoding systems use tree-based drafting (Miao et al., 2024; Cai et al., 2024; Li et al., 2024a), which may interact differently with the joint training objective.

Finally, the effect of key hyperparameters, including the regularization strength, the LK loss schedule, and the LoRA parameters, remains unexplored. Also, our training budget study covers up to 60K examples, and behavior at larger scales is not directly measured.

References

Gregor Bachmann, Sotiris Anagnostidis, Albert Pumarola, Markos Georgopoulos, Artsiom Sanakoyeu, Yuming Du, Edgar Schönfeld, Ali K. Thabet, and Jonas Kohler. 2025. Judge decoding: Faster speculative sampling requires going beyond model alignment. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Jijie Li, Li Du, Hanyu Zhao, Bo-wen Zhang, Liangdong Wang, Boyan Gao, Guang Liu, and Yonghua Lin. 2025a. Infinity instruct: Scaling instruction selection and synthesis to enhance language models. *arXiv preprint arXiv:2506.11116*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024a. Eagle-2: Faster inference of language models with dynamic draft trees. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 7421–7432.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024b. Eagle: speculative sampling requires rethinking feature uncertainty. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2025b. Eagle-3: Scaling up inference acceleration of large language models via training-time test. In *Advances in Neural Information Processing Systems*, volume 38, pages 136737–136756. Curran Associates, Inc.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, and 1 others. 2024. Specinfer: Accelerating large language model serving with tree-based speculative inference and verification. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pages 932–949.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Alexander Samarin, Sergei Krutikov, Anton Shevtsov, Sergei Skvortsov, Filipp Fisin, and Alexander Golubev. 2026. LK losses: Direct acceptance rate optimization for speculative decoding. *CoRR*, abs/2602.23881.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding

benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pages 95266–95290. Curran Associates, Inc.

Davis Wertheimer, Joshua Rosenkranz, Thomas Parnell, Sahil Suneja, Pavithra Ranganathan, Raghu Ganti, and Mudhakar Srivatsa. 2024. Accelerating production llms with combined token/embedding speculators. *arXiv preprint arXiv:2404.19124*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. 2024. Distillspec: Improving speculative decoding via knowledge distillation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen GONG, James Hoang, Armel Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, and 14 others. 2025. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, volume 2025, pages 66602–66656.