TRAINING-FREE DROPOUT SAMPLING FOR SEMANTIC TOKEN ACCEPTANCE IN SPECULATIVE DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Speculative decoding accelerates large language model inference by proposing tokens with a lightweight draft model and selectively accepting them using a target model. This work introduces DropMatch, a novel approach that matches draft tokens to the predictive distribution of the target model via Monte Carlo dropout applied exclusively to the LM head, enabling sampling-based acceptance decisions. By generating multiple decoding paths, our method forms an empirical token distribution against which draft tokens are evaluated for consistency. This acceptance mechanism enables the model to adaptively control the size of decoding paths under an appropriate dropout probability, preventing substantial distortion of the target model predictive distribution. The proposed method operates in a training-free manner, requires no architectural modification to pretrained models, and can be orthogonally integrated with a wide range of existing speculative decoding and inference acceleration techniques. Experiments across multiple benchmarks demonstrate that our approach increases acceptance length while maintaining competitive task performance, yielding inference speedups ranging from 1.09× to 1.33× over the standard baseline, and up to an additional 1.09× speedup when applied on top of EAGLE3.

1 INTRODUCTION

Large language models (LLMs) have consistently demonstrated improved performance across a wide range of tasks as model size and computational capacity increase (Grattafiori et al., 2024; Yang et al., 2025; Kaplan et al., 2020). Such scaling trends have been observed across diverse domains, including language understanding, reasoning (Wei et al., 2023), and code generation (Rozière et al., 2024), indicating that increases in model scale lead to stronger representational and generalization capabilities. However, such gains inevitably come with increased inference costs, making efficient inference with preserved model performance a central challenge for practical deployment. One of the primary bottlenecks in LLM inference lies in the autoregressive decoding process (Shazeer, 2019). Under autoregressive decoding, tokens are generated sequentially, with each token conditioned on all previously generated tokens, enforcing a strictly serial computation pattern. For example, DeepSeek-R1 (Guo et al., 2025; DeepSeek-AI et al., 2024), which has 671 billion total parameters, typically requires partitioning across multiple NVIDIA H200 GPUs to fit the model weights in memory, yet still takes several seconds to process a single inference request. Moreover, inference latency becomes even more amplified in reasoning mode (OpenAI et al., 2025) or agentic workloads (Yao et al., 2023), where the number of generated tokens increases substantially.

To address this limitation, various inference acceleration techniques such as speculative decoding have been proposed (Leviathan et al., 2023; Chen et al., 2023). Speculative decoding improves decoding efficiency by allowing a smaller draft model to propose multiple tokens in advance, which are then verified by a larger target model. A key factor in speculative decoding is how many proposed tokens can be accepted in a single verification step, as the acceptance length directly determines the overall speedup. Therefore, increasing acceptance length without sacrificing accuracy remains a central challenge in speculative decoding.

In this work, we propose DropMatch, a sampling-based acceptance method that leverages Monte Carlo (MC) dropout (Gal & Ghahramani, 2016) to improve the efficiency of speculative decoding.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Our approach applies MC dropout exclusively to the LM head of the target model, allowing multiple stochastic forward passes at the head level to produce diverse token samples without additional training. The resulting MC dropout samples are used to efficiently evaluate whether draft tokens are consistent with the target model predictions, increasing acceptance length with minimal computational overhead. As a result, DropMatch achieves practical speedups without modifying the overall speculative decoding framework.

We demonstrate the effectiveness of DropMatch across multiple model families (Grattafiori et al., 2024; Yang et al., 2025) and a broad set of reasoning (Cobbe et al., 2021), language understanding (Hendrycks et al., 2021), and instruction following (Zhou et al., 2023) benchmarks.

Overall, DropMatch consistently improves acceptance length and translates these acceptance gains into end to end decoding speedups while preserving task performance.

In summary, our contributions are as follows:

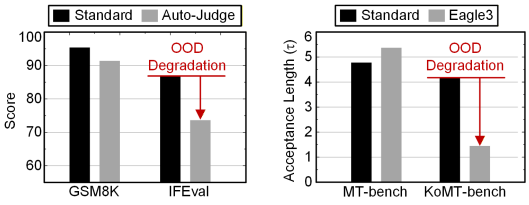
- We introduce DropMatch, a sampling based acceptance method for speculative decoding that leverages MC dropout applied only to the target model LM head. This design yields multiple token samples within a single decoding step, avoiding temperature-based heuristics and repeated full model evaluations.
- DropMatch requires no training or auxiliary data, and adds only a small additional verification cost. It consistently increases acceptance length and translates these gains into end-to-end decoding speedups, yielding improved accuracy latency trade offs.
- We demonstrate that DropMatch is compatible with existing speculative decoding pipelines rather than competing with them: combined with EAGLE-3 (Li et al., 2025c) it yields further acceleration, and combined with an external judging mechanism (Garipov et al., 2025) it gives a tunable accuracy–latency trade-off, in both cases without giving up the original advantages of those methods.

2 Related Works

Speculative Decoding Speculative decoding (Leviathan et al., 2023; Xia et al., 2023) accelerates a large target model by having a smaller draft model propose tokens that the target then verifies in a single step. Better alignment (Zhou et al., 2024; Hu et al., 2025) between the two therefore yields a higher average number of accepted tokens and greater acceleration.

Lossless Speculative Decoding Lossless speculative decoding guarantees that the final output tokens follow exactly the same sampling distribution as the target model through a strict verification process. Prior work (Li et al., 2025c; Cai et al., 2024) in this category has focused on strengthening the alignment between draft and target models while enabling faster token proposal and verification. Additionally, since a rejection at a certain position causes all subsequent tokens to be rejected, some studies (Li et al., 2025b; Huang et al., 2025; Zhang et al., 2025) have proposed using depth-specialized models to improve acceptance rates. However, in lossless approaches, even semantically equivalent tokens are rejected if they differ at the token level, which fundamentally limits the achievable speedup.

Lossy Speculative Decoding Lossy speculative decoding relaxes the strict requirement of sampling from the exact distribution of the target model, allowing tokens proposed by a draft model to be accepted as long as the output semantics or quality are preserved. A representative approach is Judge Decoding (Bachmann et al., 2025), which trains a dedicated judge head from human annotations to detect positions where a token substitution would change semantics. More recent methods reduce or remove human supervision: Auto-Judge (Garipov et al., 2025) and Self-Judge (Yoon et al., 2025) let models identify quality-critical positions autonomously, but still rely on additional learned components or auxiliary training data. This reliance can limit robustness under domain shift: when the training distribution of the judge head or the draft model is narrow, effectiveness can degrade on out-of-distribution benchmarks. Fig. 1 illustrates this effect empirically: Auto-Judge, where the judge head is trained on mathematical data, shows reduced performance on IFEval, and EAGLE-3, where the draft model is trained on English data, exhibits markedly shorter acceptance lengths on the Korean KoMT-bench (LG AI Research, 2024) benchmark.



(a) Math data (b) English data

Figure 1: Out-of-distribution results.

Figure 1a shows that Auto-Judge, using a judge head trained with Llama-3.1-70B-Instruct, preserves the acceptance length under out-of-distribution conditions while degrading task performance. Figure 1b shows that EAGLE-3, using a draft model trained with Llama-3.3-70B-Instruct, maintains task performance while the acceptance length decreases on KoMT-Bench, resulting in reduced acceleration benefits. Here, τ denotes the mean acceptance length.

The proposed method, DropMatch, falls into the category of lossy speculative decoding. When tokens are semantically similar, sampling tends to produce aligned candidates, and even when they differ, similar probability distributions can be regarded as approximating the target model. Unlike approaches that require new architectures such as EAGLE (Li et al., 2024) or POSS (Huang et al., 2025), additional data or judge heads as in Judge Decoding (Bachmann et al., 2025) or Auto-Judge (Garipov et al., 2025), or data-driven calibration (Gautam et al., 2025), our method is simple and broadly applicable to speculative decoding.

3 Methodology

In this section, we describe how DropMatch enables efficient token sampling and how the resulting samples are used to determine token acceptance. Prior speculative decoding methods are typically inductive, relying on training the draft model to closely approximate the output distribution of the target model. In contrast, our approach operates in a transductive manner at inference time, directly leveraging the predictive distribution of the target model to construct an empirical acceptance region, inspired by the k-nearest neighbor (Cover & Hart, 1967) mechanism.

3.1 Multi-Sample LM Head via MC Dropout

MC dropout has been widely used to approximate ensemble effect (Srivastava et al., 2014) and to quantify predictive uncertainty (Gal & Ghahramani, 2016) in neural networks. We revisit MC dropout as a sampling mechanism and propose a novel method, termed DropMatch, to assess whether tokens proposed by a draft model are consistent with predictions of the target model. To avoid repeated full forward passes, MC dropout is applied exclusively to the LM head, which preserves KV-cache alignment of the remaining transformer blocks. By applying MC dropout only at the LM head, this design preserves KV-cache alignment of the remaining transformer blocks, making it straightforward to implement.

For clarity, the LM head is modeled as producing K stochastic predictions by applying independent dropout masks to the final hidden representation. Let $h_t \in \mathbb{R}^d$ denote the last-layer hidden state at time step t. For each path $i \in \{1, \ldots, K\}$, a mask $m^{(i)} \in \{0,1\}^d$ is sampled with i.i.d. entries:

$$m_j^{(i)} \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(1 - p_{\text{drop}}), \quad j = 1, \ldots, d. \tag{1}$$

Using inverted-dropout scaling, the masked representation is defined as

$$h_t^{(i)} = \frac{h_t \odot m^{(i)}}{1 - p_{\text{drop}}}, \quad i = 1, \ldots, K. \tag{2}$$

Eq. equation 2 preserves the expectation $\mathbb{E}[h_t^{(i)}] = h_t$, matching the standard dropout convention. The LM head then produces the corresponding logits $l_t^{(i)}$ and token probabilities $p_t^{(i)}$ as

$$l_t^{(i)} = W h_t^{(i)}, \quad p_t^{(i)} = \text{Softmax}\left(l_t^{(i)}\right), \quad i = 1, \ldots, K. \tag{3}$$

All paths share the same LM-head weights W and differ only through the sampled dropout masks, enabling K parallel samples without introducing additional parameters.

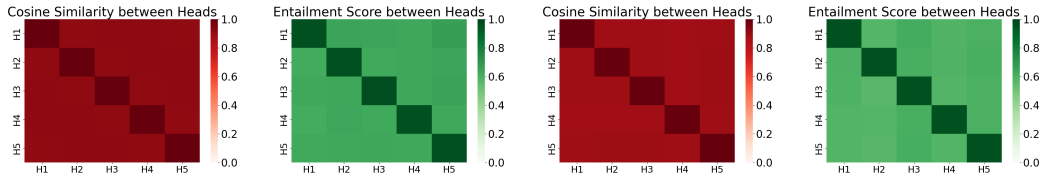


Figure 3: Semantic similarity across decoding paths. (a) Cosine similarity (Sentence-BERT) and entailment-based consistency at $p_{drop} = 0.1$, (b) results at $p_{drop} = 0.3$, showing higher similarity at lower dropout rates. H1–H5 denote MC dropout paths with $K = 5$, each from a distinct stochastic forward pass. Darker colors indicate higher values.

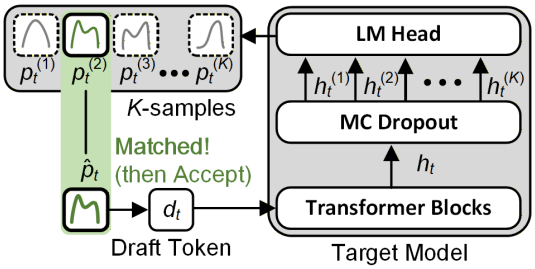


Figure 2: Overall architecture of DropMatch

Fig. 2 illustrates the overall architecture of the proposed method, which enables speculative decoding with multiple sampling via MC dropout applied at the LM head. Specifically, d_t denotes the t-th draft token, and h_t denotes its corresponding final embedding vector. We generate K stochastic representations, $h_t^{(1)}, \ldots, h_t^{(K)}$, by applying K different dropout masks to h_t, corresponding to K MC dropout paths.

As a preliminary validation, Fig. 3 shows that MC dropout paths on Llama-3.1-70B-Instruct (Grattafiori et al., 2024) stay semantically consistent on spec-bench (Xia et al., 2024): following the semantic uncertainty protocol (Kuhn et al., 2023), we measure cosine similarity between paths with sentence-BERT (Reimers & Gurevych, 2019) and entailment with a sentence-level model (He et al., 2021), and with $K = 5$ consistency increases as the dropout probability decreases, without any dropout-specific training.

Table 1 further reports the performance of individual heads on the HumanEval benchmark(Chen et al., 2021). The results show that each path produced by the LM head continues to outperform the draft model and maintains comparable accuracy when the dropout probability is moderate. Based on these observations, we demonstrate in Section 4 that aligning with at least one of multiple sampled paths during the verification phase can effectively increase acceptance length without degrading task performance.

Table 1: Pass@1 performance of Llama-3.1-70B-Instruct across different heads and dropout probabilities. Baseline Pass@1 of 81.7 without dropout; Llama-3.1-8B-Instruct used as the draft model with Pass@1 of 72.6.

L31 70B Inst	**Head1**	**Head2**	**Head3**	**Head4**	**Head5**
Pass@1 (p=0.1)	81.1	78.7	81.1	86.0	77.4
Pass@1 (p=0.2)	80.5	81.1	80.5	81.1	78.7
Pass@1 (p=0.3)	78.7	81.1	75.6	77.4	79.9
Pass@1 (p=0.4)	76.2	79.3	80.5	73.2	76.2
Pass@1 (p=0.5)	78.7	65.2	73.8	64.0	70.1

3.2 Acceptance Criteria

Standard speculative decoding determines token acceptance based on rejection sampling by comparing token probabilities from the draft and target models. In our approach, we treat the K output distributions from the multiple decoding paths as a single cluster and evaluate whether the draft token belongs to this cluster.

Naive Token-Matching Criterion The simplest criterion accepts a draft token if it matches any token produced by the K decoding paths. Let $\hat{y}_t$ denote the token proposed by the draft model at time step t, and let $y_t^{(i)} = \arg\max(p_t^{(i)})$ denote the token selected by the i-th head. The acceptance

condition is

$$\hat{y}_t \in \{y_t^{(i)} \mid i = 1, \ldots, K\}. \tag{4}$$

When the dropout heads yield sufficiently stable predictions, Eq. 4 can substantially increase the acceptance length with negligible overhead. However, because this rule ignores the full probability mass and relies only on the top-1 tokens or selected token by nucleus sampling (Holtzman et al., 2020), it can accept a draft token even when the underlying distributions are not well aligned.

JS-Divergence–Based Criterion To address this limitation, we more generally compare the draft distribution with the cluster of MC dropout head distributions using Jensen–Shannon (JS) divergence. We first define a centroid distribution by averaging the head logits and normalizing:

$$\bar{p}_t = \mathrm{Softmax}\left(\frac{1}{K}\sum_{i=1}^{K} l_t^{(i)}\right), \tag{5}$$

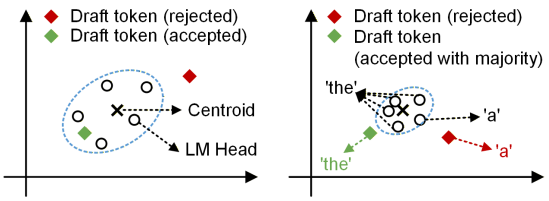
where $l_t^{(i)}$ denotes the logits produced by the i-th head. Let $\hat{p}_t$ denote the draft model distribution at time step t. We accept the draft token if its divergence to the centroid is no larger than the maximum divergence observed among the MC dropout heads:

$$\mathrm{JS}(\hat{p}_t \,\|\, \bar{p}_t) \;\le\; \max_{i=1,\ldots,K} \mathrm{JS}\left(p_t^{(i)} \,\|\, \bar{p}_t\right). \tag{6}$$

This criterion places the draft distribution inside the region the dropout paths themselves span: the centroid is its center and the largest divergence among the paths is its radius, which we refer to as the acceptance radius. When the LM head distributions are highly similar, the draft distribution may still be rejected despite being sufficiently close (Fig. 4b). Since such cases indicate that the target model has effectively collapsed to a single dominant output, we introduce an additional criterion:

$$\mathrm{majority}\left((y_t^{(i)})_{i=1}^{K}\right) = \hat{y}_t \tag{7}$$

In other words, a draft token is accepted if it matches the majority token among the multiple heads. Table 2a characterises when the majority rule is invoked. Among the positions it accepts, 98.4% have all K heads in agreement, with mean head probability 0.94 on the majority token. The rule therefore acts as a correction for the degenerate case in which the sample cluster collapses — where the acceptance radius vanishes, and Eq. 6 would reject even the majority token — rather than admitting tokens under dispersed heads(Table 10). Table 2b compares the JS divergence of tokens accepted under Eq. 6 with those accepted under Eq. 7: the latter are very close to the centroid yet fall outside the acceptance radius, which is precisely the case Eq. 6 alone cannot handle. Notably, cases accepted by Eq. 7 often exhibit very low JS divergence, yet would be rejected when relying solely on Eq. 6, highlighting the limitation of using the JS-divergence criterion alone.



(a) Dispersed samples (b) Concentrated samples

Figure 4: Conceptual illustration of the JS-divergence-based acceptance criterion.

This behavior is illustrated in Fig. 4. When MC dropout samples are dispersed, as shown in Fig. 4a, acceptance based solely on Eq. 6 can be unstable due to inconsistent predictions across heads. In contrast, when the samples are highly concentrated, as in Fig. 4b, Eq. 7 enables robust acceptance by leveraging strong agreement among heads, even in cases where Eq. 6 would reject the token. The complete acceptance procedure is summarized in Algorithm 1.

The width of this region is not a free parameter. Under inverted dropout the head logits satisfy $\mathrm{Var}\left[l_{t,v}^{(i)}\right] = \frac{p_{\text{drop}}}{1-p_{\text{drop}}}\sum_j W_{vj}^2 h_{t,j}^2$, so the acceptance radius at each position is set by the hidden state the target model actually produced there, rather than by a threshold function fixed in advance; on GSM8K it correlates with the entropy of the target distribution at $\rho = 0.924$. Replacing the dropout samples with Gaussian noise on the logits reproduces the acceptance length only within a narrow band of scales: at a quarter of that scale the gain disappears, and at four times it accuracy falls by 7.5 points (Table 12). Fixing the region instead to the top-k of the target distribution trades accuracy for acceptance length at every k we tried (Section C.1).

Table 2: Statistics of head alignment and JS-divergence behavior for the Llama-3.1-70B-Instruct model on HumanEval with MC dropout using $K = 5$ heads.

(a) Head alignment statistics

Majority	1	2	3	4	5
Eq. 7 accept	0.00%	0.14%	0.82%	0.54%	98.40%
Mean prob	0.255	0.325	0.460	0.517	0.939

(b) Average JS divergence

	$\mathrm{JS}(\bar{p}_t, \text{draft})$	$\mathrm{JS}(\bar{p}_t, \text{heads})$
Dispersed	0.0254	0.0313
Concentrated	0.0097	0.0014

Algorithm 1 JS-Divergence-Based Acceptance

Input: draft token $\hat{y}_t$, draft prob $\hat{p}_t$, LM head logits $l_t^{(i)}{}_{i=1}^{K}$, LM head probs $p_t^{(i)}{}_{i=1}^{K}$
Compute centroid: $\bar{p}_t = \mathrm{Softmax}\left(\frac{1}{K}\sum_{i=1}^{K} l_t^{(i)}\right)$
Compute LM head tokens: $y_t^{(i)} = \arg\max p_t^{(i)}$
if $JS(\hat{p}_t||\bar{p}_t) \leq \max_i JS(p_t^{(i)}||\bar{p}_t)$ **then**
  Accept
else if $\mathrm{majority}\left(\{y_t^{(i)}\}_{i=1}^{K}\right) = \hat{y}_t$ **then**
  Accept
else
  Reject and use $\mathrm{majority}\left((y_t^{(i)})_{i=1}^{K}\right)$ as the next token
end if

4 Experiments

We evaluate DropMatch in terms of (i) computational overhead, (ii) acceptance rate, and (iii) end-to-end decoding speed. We first quantify the cost of MC dropout at the LM head (Section 4.1), then apply the method to speculative decoding with Llama-3.1 and Qwen3 across GSM8K, MMLU, IFEval, and HumanEval (Section 4.2). Finally, we demonstrate its complementarity with prior approaches by integrating it with Auto-Judge and EAGLE-3 (Section 4.3–Section 4.4).

Metrics

- **Accuracy**: Since the proposed method does not strictly sample from the exact target model distribution, we report accuracy to assess potential performance degradation.
- **Mean Acceptance Length** (τ): Average tokens per step, including accepted draft tokens and the target bonus token (up to $L + 1$). Higher τ yields greater speedup. Draft length is L.
- **Throughput / Speedup**: We report decoding speed in tokens per second (tokens/s) and relative speedup over the baseline model.

4.1 Overhead of MC dropout sampling in the LM head

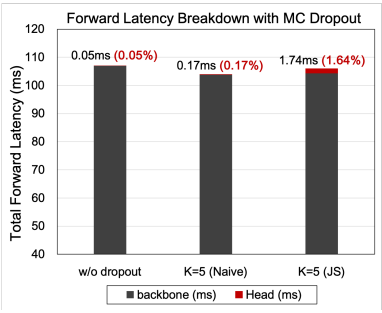


Figure 5: Forward latency overhead

We first demonstrate that applying MC dropout to the LM head incurs only minimal computational overhead. Specifically, we analyze the inference time of the Llama-3.1-70B-Instruct model by decomposing it into transformer blocks and the LM head, and comparing performance with and without MC dropout (Figure 5). As observed in recent work (Lu et al., 2025), the computational cost of the LM head is negligible compared to the overall inference cost. Consistent with this observation, our measurements show that the LM head accounts for only 0.05% of the total forward cost, and that even with five MC dropout paths and JS divergence computation, the total overhead remains at 1.64%. Based on this observation, we show that selectively using Naive Token-Matching and

JS divergence-based criteria allows us to achieve effective speedups while minimizing additional overhead.

4.2 SPECULATIVE DECODING WITH DROPMATCH

For fair comparison, we evaluate speculative decoding with DropMatch using the vLLM (Kwon et al., 2023), lm-evaluation-harness (Gao et al., 2024) and EvalPlus (Liu et al., 2023) frameworks. In these experiments, we employ the JS divergence–based acceptance criterion and report results with a batch size of 1 for throughput or speed up. Unless otherwise specified, we use $K = 5$ and $p_{\text{drop}} = 0.3$, which are selected based on the ablation study in Section A.

Table 4 summarizes the experimental results on the Llama-3.1-8B/70B-Instruct and Qwen3 4B/32B models. Both model pairs were evaluated using 4 A100-SXM4-80G GPUs in tensor-parallel mode. Across most tasks, we observe approximately a 10% increase in acceptance length on draft length $L = 5$, which corresponds to a similar improvement in decoding speed over standard speculative decoding. These gains are achieved at almost no additional cost, with little to no degradation in accuracy. Compared to standard speculative decoding, DropMatch achieves up to a 1.33× throughput improvement on the Qwen3 4B/32B models at batch size 1 with draft length $L = 10$. For the Llama-3.1-8B/70B-Instruct models, DropMatch yields a 1.19× speedup on GSM8K under the same $L = 10$ setting. In contrast, HumanEval leaves little headroom: across all verification positions the K heads are unanimous 87.5% of the time, compared with 74.7% on GSM8K (Table 9). The acceptance region therefore coincides with the greedy one at most positions, and the standard model already exhibits a high acceptance rate, making further increases in acceptance length difficult — particularly under the strict syntactic requirements of code generation. Table 3 reports relative throughput improvements for the Llama-3.1-70B-Instruct model, as shown in Table 4, improvement is consistently maintained at approximately 1.10× even when the batch size is increased to 128.

Table 3: Throughput improvement across batch sizes for the GSM8K with draft length $L = 10$ on A100-SXM4-80G GPU with Llama3.1-70B-Instruct. Standard speculative decoding(SD) as the baseline (1.0×), both SD and DropMatch(DM) were measured using vLLM with tensor parallelism TP=4.

Batch size	1	4	16	32	64	128
SD + DM	1.19x	1.18x	1.12x	1.10x	1.15x	1.10x

4.3 AUTO-JUDGE WITH DROPMATCH

Following Section 4.1, we apply DropMatch to the Auto-Judge framework. In this experiment, we use the Llama-3.1-70B-Instruct model and evaluate performance on the GSM8K and LiveCodeBench (Jain et al., 2024) datasets, following the experimental setup of Auto-Judge. We reproduce the Auto-Judge experiments by training judge heads using the provided training datasets and the official repository[1] code for both GSM8K and LiveCodeBench. Fig. 6a shows that on GSM8K, under identical Auto-Judge parameter settings, the proposed method maintains comparable accuracy while achieving longer mean acceptance length than the baseline Auto-Judge approach. Table 6 reports results on both GSM8K and LiveCodeBench, including mean acceptance length and accuracy across different judge-head threshold settings. Under the Auto-Judge inference setup with batch size set to 1, the proposed method consistently achieves longer acceptance lengths than the baseline while maintaining comparable accuracy across all threshold configurations. This trend holds across both datasets, demonstrating the robustness of our approach when combined with Auto-Judge.

4.4 EAGLE-3 WITH DROPMATCH

For experiments with EAGLE-3, we adopt the Naive Token-Matching criterion instead of the JS divergence–based acceptance rule. EAGLE-3 models employ tree decoding, where candidate tokens are sampled from the same probability distribution, making JS divergence–based evaluation redundant and computationally inefficient due to repeated calculations. In contrast, the Naive Token-Matching

[1] https://github.com/garipovroma/autojudge

Table 4: Performance comparison of standard speculative decoding(SD) and its combination with DropMatch(DM) on Llama-3.1 and Qwen3 models. Accuracy, mean acceptance length, and throughput on GSM8K, MMLU, IFEval, and HumanEval benchmarks.

Model	Method	L	GSM8K			MMLU			IFEval			HumanEval		
			Speedup	τ	Acc.	Speedup	τ	Acc.	Speedup	τ	Acc.	Speedup	τ	Pass@1
Llama-3.1 8B/70B-Instruct	SD	5	1.0x	4.97	94.69	1.0x	4.02	85.94	1.0x	4.12	85.60	1.0x	5.40	81.70
	SD + DM	5	1.10x	5.50	94.29	**1.09x**	4.43	85.50	**1.10x**	4.48	85.24	1.00x	5.57	79.89
	SD	10	1.02x	7.62	95.40	0.86x	5.25	86.06	0.87x	5.47	86.86	**1.13x**	8.87	81.70
	SD + DM	10	**1.22x**	9.20	93.90	1.02x	6.06	85.34	0.99x	6.34	86.17	1.12x	9.38	78.70
Qwen3 4B/32B	SD	5	1.0x	5.06	86.20	1.0x	4.62	83.21	1.0x	3.40	80.72	1.0x	5.09	90.4
	SD + DM	5	1.07x	5.66	87.49	1.05x	5.15	82.15	**1.19x**	4.22	78.24	1.00x	5.27	90.9
	SD	10	0.96x	7.87	86.20	1.12x	6.48	83.26	0.72x	4.04	82.31	1.31x	8.11	92.1
	SD + DM	10	**1.13x**	9.78	86.58	**1.49x**	8.60	82.06	0.98x	5.67	81.60	**1.39x**	8.72	90.2

Table 5: Performance comparison of EAGLE-3 and EAGLE-3 + DropMatch(DM) on GSM8K, MT-bench, Alpaca with the Llama-3.3-70B-Instruct EAGLE-3 model across different draft lengths. Speedups are reported relative to the standalone baseline (1.0×).

Model	Method	L	GSM8K			MT-bench			Alpaca		
			Speedup	τ	Acc.	Speedup	τ	Score	Speedup	τ	Win rate
Llama-3.3 70B-Instr.	EAGLE-3	5	4.19x	5.82	78.75	3.90x	5.37	8.47	4.31x	5.83	50.0
		7	4.65x	6.71	78.75	4.18x	5.81	8.43	4.75x	6.59	51.3
		9	4.68x	6.71	78.75	4.24x	6.06	8.48	4.92x	6.89	50.6
	EAGLE-3 +DM	5	4.32x	6.09	78.75	4.13x	5.86	8.45	4.47x	6.30	55.0
		7	4.86x	7.07	81.25	4.52x	6.58	8.46	5.02x	7.25	56.0
		9	**5.03x**	**7.48**	80.00	**4.62x**	**6.89**	8.59	**5.27x**	**7.74**	52.0

criterion introduces no additional divergence computation overhead and can be applied even when the vocabularies of the draft and target models differ, enabling broader applicability. In addition, to ensure faithful reproduction of EAGLE-3, we conduct all experiments using the official EAGLE-3 repository[2] code.

Table 5 presents results for Llama-3.3-70B-Instruct EAGLE-3 combined with the proposed method, showing that acceptance length and decoding speed improvements are preserved. While increasing draft length in EAGLE-3 alone leads to saturation in acceptance length and speed, combining it with our method enables additional acceleration without significant performance degradation. Experiments follow the EAGLE-3 setting with 80 samples and varying draft lengths. GSM8K provides ground-truth answers evaluated by exact match, whereas MT-Bench (Zheng et al., 2023) and Alpaca (Taori et al., 2023) adopt an LLM-as-a-Judge (Li et al., 2025a) paradigm. MT-Bench reports scores on a 10 point scale, and Alpaca uses win rates against standard speculative decoding. On the full GSM8K dataset, the standard model and EAGLE-3 achieve 81.7%, while EAGLE-3+DropMatch attains 81.3% with $L = 10$, indicating preserved performance. For MT-Bench and Alpaca, evaluated using GPT-4 (OpenAI et al., 2024), EAGLE-3 scores 8.64 and achieves a 50% win rate against standard. Applying DropMatch yields 8.52 and a 48% win rate with $L = 10$, demonstrating additional acceleration with minimal quality degradation. Across all settings, EAGLE-3+DropMatch consistently improves speed and extends acceptance length. Notably, on GSM8K, where EAGLE-3 alone saturates at larger draft lengths, our method further increases acceptance length while largely preserving task performance.

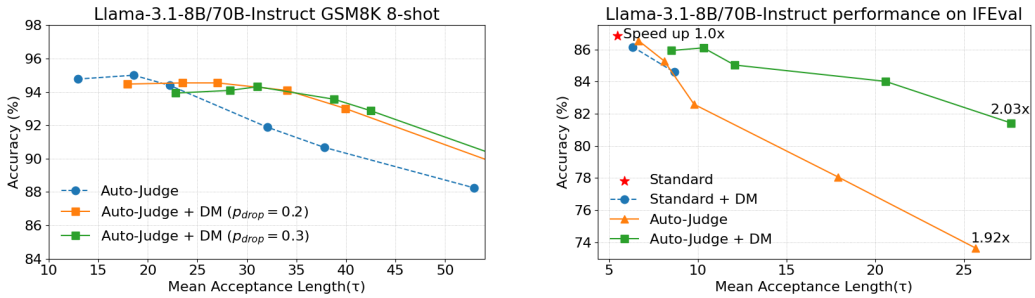
4.5 Out of Distribution Performance

Finally, we evaluate learning-based speculative decoding methods and our proposed approach on out-of-distribution (OOD) data, i.e., data that the methods have not been trained on. Since our approach does not modify the parameters of either the draft or the target model, it avoids catastrophic forgetting and mitigates degradation under distribution shifts. Fig. 6b reports results on the IFEval benchmark using Auto-Judge, whose judge head is trained on mathematical data as described in Section 4.3. Although Auto-Judge continues to achieve a substantially longer mean acceptance length than standard speculative decoding, its task performance degrades rapidly as the distribution shifts.

[2]https://github.com/SafeAILab/EAGLE

Table 6: Auto-Judge + DropMatch results on GSM8K and LiveCodeBench across thresholds.

Method	GSM8K 8-shot					LiveCodeBench				
	Metric	threshold				Metric	threshold			
		0.014	0.022	0.112	0.281		0.012	0.026	0.051	0.121
Auto-Judge	Accuracy (%)	95.1	94.3	91.4	88.2	Pass@1	29.1	28.0	28.1	24.0
	tokens/s	57.4	62.2	84.8	90.1	τ	14.1	19.0	27.5	47.0
Auto-Judge + DM	Accuracy (%)	94.1	93.9	93.0	**90.7**	Pass@1	29.4	29.1	26.8	22.4
	tokens/s	73.9	77.6	90.4	**95.9**	τ	18.6	24.0	32.7	49.5



(a) GSM8K (8-shot)

(b) IFEval (out-of-distribution)

Figure 6: Comparison of Auto-Judge and Auto-Judge combined with DropMatch (DM).

In contrast, our method maintains stable performance even when the draft length is increased, while avoiding excessively high acceptance rates. Moreover, when combined with Auto-Judge, DropMatch achieves a longer mean acceptance length under the same experimental settings, while exhibiting a more gradual degradation in performance. This trend is also observed in Fig. 6a, suggesting that Auto-Judge frequently rejects tokens that should be accepted to improve performance, or conversely accepts tokens that should be rejected in out-of-distribution cases. By complementing these failure modes, DropMatch effectively extends acceptance length while mitigating performance degradation.

Fig. 1b presents results on the KoMT-Bench benchmark using EAGLE-3 with a draft length of $L = 5$, where the draft model is trained on English data. Under this setting, EAGLE-3 exhibits a low mean acceptance length, indicating that the target model struggles to accept tokens proposed by the draft model under distribution shift.

When increasing the draft length to $L = 7$ on KoMT-Bench, we observe a similar trend: EAGLE-3 achieves a score of 7.96 with $\tau = 1.46$, whereas standard speculative decoding achieves 8.08 with $\tau = 4.85$. Our method (SD + DM) further improves both metrics to 8.12 and $\tau = 5.24$, demonstrating consistent gains even under distribution shift.

These results suggest that simply increasing the draft length does not alleviate the mismatch between the draft and target distributions in EAGLE-3. In contrast, our approach remains robust, effectively mitigating out-of-distribution effects and enabling off-the-shelf applicability without additional training or parameter updates. results on out-of-distribution (OOD) settings are provided in Table 16.

5 Conclusion

In this work, we propose DropMatch, a novel approach to accelerate speculative decoding by applying Monte Carlo (MC) dropout exclusively to the LM head of the target model for sampling-based acceptance. The method operates without training, making it less susceptible to out-of-distribution issues. It requires no modification to pretrained model architectures and introduces negligible computational overhead. Experimental results show inference speedups of 1.09× to 1.33× over standard speculative decoding. Moreover, DropMatch is orthogonal to existing acceleration methods and can be seamlessly combined with them, yielding consistent additional speedups across diverse settings.

AI USE STATEMENT

Generative AI tools were used in two limited capacities in this work. First, they assisted with language editing to improve the grammar, clarity, and phrasing of author-written text. Second, they provided coding assistance during research execution, specifically to help modify and run experimental code. All AI-assisted code was reviewed, verified, and tested for correctness by the authors, and all reported results were produced and validated by the authors. We did not use generative AI tools for research ideation, methodology design, generating synthetic datasets, proving mathematical claims, or retrieval and discovery of related work. We have reviewed all AI-assisted work, and the authors take full responsibility for the final content of this work, including all text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

DropMatch requires no training, no auxiliary data, and no architectural change, so reproducing it amounts to modifying the verification step of an existing speculative decoding implementation. Everything needed to write that modification is in the paper: the dropout paths in Eq. 2 and Eq. 3, the centroid in Eq. 5, the JS-divergence criterion and the acceptance radius in Eq. 6, the majority criterion in Eq. 7, and the complete procedure including the fallback token in Algorithm 1. The method has two hyperparameters, K and p_{drop}; we use $K = 5$ and $p_{\text{drop}} = 0.3$ throughout unless stated otherwise, and Section A reports the ablation from which they were chosen. Our reported numbers come from implementing Algorithm 1 against a standard transformer decoding loop and porting it into vLLM 0.8.4, whose speculative decoding verification step it replaces; the Auto-Judge comparisons additionally apply the judge head released by the authors of that method through their published interface. Section 4.2 names the evaluation harnesses, the decoding temperature, and the batch sizes used for every reported number, and all benchmarks are public and are evaluated through their standard harnesses without modification. Section B and Section C state the measurement protocol, sample sizes, and statistical tests for the analyses reported in the appendix. We will release the full source code upon acceptance.

REFERENCES

Gregor Bachmann, Sotiris Anagnostidis, Albert Pumarola, Markos Georgopoulos, Artsiom Sanakoyeu, Yuming Du, Edgar Schönfeld, Ali Thabet, and Jonas K Kohler. Judge decoding: Faster speculative sampling requires going beyond model alignment. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mtSSFiqW6y.

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

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling, 2023. URL https://arxiv.org/abs/2302.01318.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John

Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Thomas M. Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13(1):21–27, 1967. URL http://dblp.uni-trier.de/db/journals/tit/tit13.html#CoverH67.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. DeepSeek-V3 Technical Report, December 2024. URL http://arxiv.org/abs/2412.19437. arXiv:2412.19437 [cs].

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Roman Garipov, Fedor Velikonivtsev, Ruslan Svirschevski, Vage Egiazarian, and Max Ryabinin. Autojudge: Judge decoding without manual annotation, 2025. URL https://arxiv.org/abs/2504.20039.

Aayush Gautam, Susav Shrestha, and Narasimha Reddy. Token-driven gammatune: Adaptive calibration for enhanced speculative decoding, 2025. URL https://arxiv.org/abs/2504.00030.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru,

Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik

Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

Shijing Hu, Jingyang Li, Xingyu Xie, Zhihui Lu, Kim-Chuan Toh, and Pan Zhou. Griffin: Effective token alignment for faster speculative decoding, 2025. URL https://arxiv.org/abs/2502.11018.

Langlin Huang, Chengsong Huang, Jixuan Leng, Di Huang, and Jiaxin Huang. Poss: Position specialist generates better draft for speculative decoding, 2025. URL https://arxiv.org/abs/2506.03566.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation, 2023. URL https://arxiv.org/abs/2302.09664.

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

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

LG AI Research. KoMT-Bench. https://huggingface.co/datasets/LGAI-EXAONE/KoMT-Bench, 2024.

Dawei Li, Zhen Tan, Chengshuai Zhao, Bohan Jiang, Baixiang Huang, Pingchuan Ma, Abdullah Alnaibari, Kai Shu, and Huan Liu. Who's your judge? on the detectability of llm-generated judgments, 2025a. URL https://arxiv.org/abs/2509.25154.

Jinze Li, Yixing Xu, Haiduo Huang, Xuanwu Yin, Dong Li, Edith C. H. Ngai, and Emad Barsoum. Gumiho: A hybrid architecture to prioritize early tokens in speculative decoding, 2025b. URL https://arxiv.org/abs/2503.10135.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *International Conference on Machine Learning*, 2024.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-3: Scaling up inference acceleration of large language models via training-time test. In *Annual Conference on Neural Information Processing Systems*, 2025c.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1qvx610Cu7.

Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Wei Liu, Jian Luan, Xiwen Zhang, Nicholas D. Lane, and Mengwei Xu. Demystifying small language models for edge deployment. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14747–14764, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.718. URL https://aclanthology.org/2025.acl-long.718/.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives,

Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024. URL https://arxiv.org/abs/2308.12950.

Noam Shazeer. Fast transformer decoding: One write-head is all you need, 2019. URL https://arxiv.org/abs/1911.02150.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language models are multilingual chain-of-thought reasoners, 2022. URL https://arxiv.org/abs/2210.03057.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014. URL jmlr.org.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation, 2023. URL https://arxiv.org/abs/2203.16487.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 7655–7671, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.456. URL https://aclanthology.org/2024.findings-acl.456.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Kanghoon Yoon, Minsub Kim, Sungjae Lee, Joonhyung Lee, Sunghyeon Woo, Yeonjun In, Se Jung Kwon, Chanyoung Park, and Dongsoo Lee. Selfjudge: Faster speculative decoding via self-supervised judge verification, 2025. URL https://arxiv.org/abs/2510.02329.

Lefan Zhang, Xiaodan Wang, Yanhua Huang, and Ruiwen Xu. Learning harmonized representations for speculative sampling. In *International Conference on Learning Representations*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. Distillspec: Improving speculative decoding via knowledge distillation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rsY6J3ZaTF.

A Ablation Study on K and p_{DROP}

In this section, we provide additional ablation studies on the number of Monte Carlo dropout paths K and the dropout probability p_{drop}, using Llama-3.1-8B-Instruct as the draft model and Llama-3.1-70B-Instruct as the target model. We analyze how these hyperparameters affect both task performance and acceptance length across different settings.

A.1 Effect of K

To investigate the effect of K, we fix p_{drop} and conduct an ablation study with respect to K. As shown in Table 7, when $p_{\text{drop}} = 0.3$, we find that $K = 5$ yields the most favorable performance. We further verify through experiments in the main text that this choice generalizes well across a wide range of tasks.

Table 7: Ablation study on the number of dropout paths K with $p_{\text{drop}} = 0.3$ on GSM8K. We report accuracy, mean acceptance length (τ), and speedup.

K	p_{drop}	**GSM8K (%)**	τ	**Speedup**
1	0.3	89.01	3.988	0.83x
2	0.3	92.52	4.560	0.92x
3	0.3	93.68	5.184	1.03x
4	0.3	93.73	5.328	1.05x
5	0.3	**94.29**	5.503	**1.10x**
6	0.3	94.13	5.595	1.10x
7	0.3	94.16	5.634	1.10x
8	0.3	92.86	5.562	1.09x
9	0.3	93.63	5.658	1.10x

A.2 Effect of p_{DROP}

For p_{drop}, we fix K and perform an ablation study. As reported in Table 8, the suitable range for p_{drop} lies within $[0.2, 0.4]$. In our main experiments, we primarily adopt $p_{\text{drop}} = 0.3$, as it provides a favorable trade-off between accuracy and speedup.

Table 8: Ablation study on dropout probability p_{drop} with $K = 5$ on GSM8K. We report accuracy, mean acceptance length (τ), and speedup.

K	p_{drop}	**GSM8K (%)**	τ	**Speedup**
5	0.1	93.71	5.112	1.02x
5	0.2	**94.54**	5.292	1.05x
5	0.3	94.29	5.503	1.10x
5	0.4	93.56	5.604	1.11x
5	0.5	92.87	5.802	1.14x
5	0.6	90.75	5.868	1.15x
5	0.7	88.93	5.952	1.16x
5	0.8	86.66	5.916	1.16x
5	0.9	86.82	6.000	**1.17x**

B Head Agreement Statistics

Table 2a is measured over the verification positions that are accepted under Eq. 7, so it characterises the state in which the majority criterion fires rather than the behaviour of the dropout heads over decoding as a whole. Because Eq. 7 is reached only after Eq. 6 has already rejected the draft token, those positions are concentrated by construction, which is why agreement there is close to unanimous. This section reports the complementary measurement over all verification positions, which is what determines how much room the dropout heads have to act on a given task. Both measurements use

Llama-3.1-8B-Instruct as the draft model and Llama-3.1-70B-Instruct as the target model with $K = 5$ and $p_{\text{drop}} = 0.3$, over 15,624 (GSM8K) and 46,514 (HumanEval) verification positions at $L = 5$.

Table 9: Head alignment over all verification positions. Rows labelled "All positions" give the fraction of verification positions at which the indicated number of heads agree with the majority token, in contrast to Table 2a, which is restricted to the positions accepted under Eq. 7. "Target prob" is the probability that the target model assigns to the majority token without dropout, and is therefore a different quantity from "Mean prob" in Table 2a, which averages the probability assigned by the agreeing heads.

	Majority	1	2	3	4	5
GSM8K	All positions ($L = 5$)	0.01%	2.50%	11.86%	10.97%	74.67%
	All positions ($L = 10$)	0.02%	2.59%	11.90%	11.14%	74.35%
	Target prob	0.430	0.270	0.489	0.639	0.961
HumanEval	All positions ($L = 5$)	0.03%	0.98%	5.13%	6.37%	87.50%
	All positions ($L = 10$)	0.03%	1.01%	5.21%	6.47%	87.27%
	Target prob	0.120	0.304	0.517	0.685	0.981

Table 9 supports two observations. First, the distribution is essentially unchanged between $L = 5$ and $L = 10$, so head agreement reflects a property of the task rather than of the draft length; this is consistent with the acceptance length gains in Table 4 scaling with the task rather than with L. Second, for every level of agreement above one head, the probability the target model assigns to the majority token increases with the number of agreeing heads, from 0.270 to 0.961 on GSM8K and from 0.304 to 0.981 on HumanEval. The number of agreeing heads therefore tracks the confidence of the target model without requiring any training or any access to the target distribution at the time the heads are formed. The single-head column covers 0.01–0.03% of positions and is too small to be read as part of this trend.

Table 10: Behaviour of the acceptance criteria at concentrated and dispersed verification positions, with $L = 5$. Positions are called concentrated when all $K = 5$ heads agree on the majority token and dispersed otherwise, following Fig. 4a and Fig. 4b. "Substituted" is the fraction of positions at which a draft token is accepted even though it differs from the token the target model would emit without dropout, and "Acceptance radius" is the median of the right-hand side of Eq. 6.

	Position type	Share	Substituted	Rejected	Acceptance radius
GSM8K	Concentrated	74.67%	0.22%	1.84%	0.0001
	Dispersed	25.33%	16.68%	16.57%	0.1695
HumanEval	Concentrated	87.50%	0.09%	1.57%	0.0000
	Dispersed	12.50%	16.45%	17.56%	0.1732

Table 10 separates the two situations illustrated in Fig. 4. At concentrated positions the maximum divergence among the MC dropout heads collapses to near zero, Eq. 6 accepts almost nothing beyond the token the target model would emit on its own, and Eq. 7 supplies the acceptance instead; this is the degenerate case the majority criterion was introduced to handle. At dispersed positions the criteria are selective rather than permissive: about one sixth of such positions are accepted with a substituted token and about one sixth are rejected outright, and these proportions are close to identical on the two tasks. The mechanism therefore behaves the same way wherever it is invoked, and what differs between tasks is only how often it is invoked, 25.33% of positions on GSM8K against 12.50% on HumanEval. This is the measurement behind the discussion in Section 4.2: HumanEval leaves roughly half as many dispersed positions for the dropout heads to act on, which limits the additional acceptance length available there even though standard speculative decoding is already strong on that task.

C Alternatives to the Sampling-Based Acceptance Region

The acceptance region of Eq. 6 is built from MC dropout samples. Two simpler constructions suggest themselves: fixing the region to the top-k of the target distribution, and keeping a stochastic region but drawing it from a perturbation that does not follow the hidden state. This section evaluates both.

All runs use Llama-3.1-8B-Instruct as the draft model, Llama-3.1-70B-Instruct as the target model, $K = 5$, $p_{\text{drop}} = 0.3$ and $L = 5$, over all 1319 GSM8K questions, with every criterion evaluated on the same questions through the same pipeline so that the comparison is paired at the question level.

C.1 Deterministic Relaxation

Table 11: Deterministic top-k acceptance against DropMatch. Fidelity is the fraction of final answers matching lossless greedy decoding, and the last column gives the discordant pairs and paired McNemar p-value against it. Top-k is deterministic and reported from a single run; DropMatch is reported as mean $\pm$ standard deviation over five runs, with the test taken from the median run.

Criterion	GSM8K (%)	τ	Fidelity (%)	vs. lossless greedy
Lossless greedy (top-1)	94.8	4.97	100.0	—
Top-2	93.1	5.71	92.0	24:43, $p = 0.027$
Top-3	90.8	5.86	90.4	17:65, $p < 0.001$
DropMatch	93.9 ± 0.2	5.47 ± 0.04	91.8 ± 1.8	20:26, $p = 0.461$

Table 11 shows that enlarging the acceptance region by a fixed amount buys acceptance length at a monotone cost in accuracy: going from top-1 to top-2 to top-3 raises τ from 4.97 to 5.86 while accuracy falls from 94.8% to 90.8%, and both steps differ significantly from lossless greedy decoding. DropMatch reaches $\tau = 5.47$ at an accuracy that is indistinguishable from it. The difference is not that the sampling-based region is uniformly wider or narrower, but that its width is decided separately at each position, so it can be wide where the target model is uncertain without also being wide where the target model is committed.

C.2 Perturbation Scale

Widening the acceptance region does not by itself explain the result. Replacing the dropout paths with Gaussian perturbations of the logits reproduces DropMatch when the scale matches the spread the dropout paths produce, but the usable range is narrow: at a quarter of that scale the acceptance length falls back to the lossless value, and at four times it the rule accepts almost every draft token and loses 7.5 accuracy points. What MC dropout contributes is therefore not the perturbation but its magnitude, which the hidden state fixes at each position rather than having to be searched for.

Table 12: Gaussian perturbation of the target logits in place of MC dropout. The acceptance rule is unchanged; only the source of the dispersion among the K paths differs. Measured over 8050 verification positions, the dropout paths spread the logits with a standard deviation of 1.39, so $\sigma = 1.4$ matches what MC dropout produces. The last column gives the discordant pairs and paired McNemar p-value against lossless greedy decoding.

Perturbation	GSM8K (%)	τ	vs. lossless greedy
None (lossless greedy)	94.8	4.97	—
MC dropout (DropMatch)	93.9 ± 0.2	5.47 ± 0.04	20:26, $p = 0.461$
Gaussian, $\sigma = 0.35$ ($\frac{1}{4}\times$)	94.5	5.02	12:15, $p = 0.701$
Gaussian, $\sigma = 1.4$ ($1\times$)	94.0	5.51	19:29, $p = 0.193$
Gaussian, $\sigma = 5.7$ ($4\times$)	87.3	5.93	20:118, $p < 0.001$

D Additional Out-of-Distribution Experiments

In this section, we present additional experiments evaluating the behavior of Auto-Judge, EAGLE-3 and DropMatch under out-of-distribution (OOD) settings.

D.1 Experimental Setup

Following the experimental protocol in the main text and Section A, we conduct experiments on the MGSM (Shi et al., 2022) dataset using 8-shot benchmarking setup, $K = 5$ and $p_{\text{drop}} = 0.3$. For

SD, SD+DM, and Auto-Judge, we use Llama-3.1-8B-Instruct as the draft model and Llama-3.1-70B-Instruct as the target model. For EAGLE-3, we use its dedicated draft model and Llama-3.3 as the target model.

D.2 Out-of-Distribution Evaluation of Auto-Judge

In this subsection, we evaluate the out-of-distribution (OOD) performance of Auto-Judge and Auto-Judge combined with DropMatch (DM) on multilingual GSM (MGSM) benchmarks. We report accuracy, mean acceptance length (τ), and speedup relative to standard speculative decoding (SD).

Table 13: Out-of-distribution results of Auto-Judge on MGSM benchmarks. Speedup is measured relative to standard speculative decoding (SD).

(a) MGSM-ZH

Method	L	τ	Acc.	Speedup
SD	5	4.806	83.2	1.0x
SD + DM	5	5.316	78.4	1.07x
SD + DM	32	15.939	78.4	0.91x
Auto-Judge	32	30.690	66.8	1.63x
Auto-Judge + DM	32	31.713	67.2	1.63x

(b) MGSM-TH

Method	L	τ	Acc.	Speedup
SD	5	4.884	64.0	1.0x
SD + DM	5	5.118	69.6	1.02x
SD + DM	32	13.794	70.8	0.78x
Auto-Judge	32	31.647	43.6	1.61x
Auto-Judge + DM	32	31.911	66.4	1.61x

(c) MGSM-DE

Method	L	τ	Acc.	Speedup
SD	5	4.872	84.8	1.0x
SD + DM	5	5.634	79.2	1.10x
SD + DM	32	21.351	82.0	1.15x
Auto-Judge	32	32.043	70.4	1.64x
Auto-Judge + DM	32	32.472	73.6	1.63x

(d) MGSM-FR

Method	L	τ	Acc.	Speedup
SD	5	4.842	82.0	1.0x
SD + DM	5	5.622	78.0	1.07x
SD + DM	32	20.460	78.0	0.91x
Auto-Judge	32	29.601	68.8	1.63x
Auto-Judge + DM	32	30.459	71.6	1.63x

From the results in Table 13, the performance degradation of SD+DM is more pronounced than in the English setting reported in the main text. To investigate the cause, we sample 12 examples per language and generate 64 tokens for each, comparing the top-1 agreement between the draft and target models. The results are summarized in Table 14.

We observe that, in non-English languages, disagreements between the draft and target occur $1.6\times$ to $2.1\times$ more frequently, indicating a greater degree of distribution shift. Furthermore, the target distribution tends to be flatter for non-English inputs. As shown in Table 15, the top-20 entropy for English is 0.26 nats, whereas other languages range from 0.37 to 0.48 nats.

This increased flatness makes the argmax more susceptible to changes under dropout, leading to more frequent mismatches. Nevertheless, compared to lossy approaches such as Auto-Judge, our method avoids accepting tokens under large distribution discrepancies, thereby better preserving accuracy.

D.3 Out-of-Distribution Evaluation of EAGLE-3

In this subsection, we evaluate EAGLE-3 under out-of-distribution (OOD) settings and compare it with speculative decoding (SD) and SD with DropMatch (DM). Since the official EAGLE-3

Table 14: Relative increase in disagreement rates compared to English.

Language	en	de	fr	zh	th
Disagreement Rate	6.77%	10.68% (1.58×)	14.06% (2.08×)	13.30% (1.96×)	10.94% (1.62×)

Table 15: Flatness of the target distribution across languages.

	en	de	fr	zh	th
Top-1 Mass	0.9005	0.8807	0.8664	0.8398	0.8748
Top-20 Entropy (nats)	0.2563	0.3718	0.3866	0.4820	0.3747

implementation does not provide strict accuracy evaluation, we report accuracy using a flexible matching criterion (flex).

Table 16: Out-of-distribution results of EAGLE-3 on MGSM benchmarks. Speedup is measured relative to the target 70B model. Accuracy is reported using flexible matching (flex).

(a) MGSM-ZH

Method	L	τ	Acc.	Speedup
SD	5	4.806	83.2	1.96x
SD + DM	5	5.316	78.4	2.09x
EAGLE-3	5	1.460	83.2 (flex)	1.15x
EAGLE-3 + DM	5	1.523	81.2 (flex)	1.15x

(b) MGSM-TH

Method	L	τ	Acc.	Speedup
SD	5	4.884	64.0	2.01x
SD + DM	5	5.118	69.6	2.05x
EAGLE-3	5	1.358	85.6 (flex)	1.08x
EAGLE-3 + DM	5	1.410	84.4 (flex)	1.09x

(c) MGSM-DE

Method	L	τ	Acc.	Speedup
SD	5	4.872	84.8	1.99x
SD + DM	5	5.634	79.2	2.19x
EAGLE-3	5	2.404	84.0 (flex)	1.87x
EAGLE-3 + DM	5	2.609	81.6 (flex)	1.94x

(d) MGSM-FR

Method	L	τ	Acc.	Speedup
SD	5	4.842	82.0	1.98x
SD + DM	5	5.622	78.0	2.23x
EAGLE-3	5	2.780	80.8 (flex)	2.13x
EAGLE-3 + DM	5	3.057	81.2 (flex)	2.26x

As discussed in Section D.2, Table 16 also shows that DM exhibits weaker draft–target agreement on non-English tasks. While a trade-off between acceleration and agreement still exists, EAGLE-3 often fails to achieve meaningful speculative decoding gains on certain tasks, even when accuracy is preserved. Nevertheless, for tasks such as French, where both acceleration and accuracy can be maintained, applying DM provides additional speedup without degrading performance.