A METHOD-OF-TYPES INTERPRETATION OF TOKEN DISTRIBUTION CONVERGENCE FOR DENOISING STEP REDUCTION IN UNIFORM-STATE DIFFUSION LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion LLMs (dLLMs) alleviate the memory bottleneck of autoregressive LLMs through block-wise parallel generation, but their generation speed remains limited by the number of denoising steps. Existing samplers either ignore convergence across consecutive steps or employ conservative criteria, leaving already-converged positions unexploited and causing redundant steps. We introduce the **Convergence Probability**, which estimates the likelihood that consecutive-step probability distributions have converged through a probabilistic interpretation based on the method of types in Shannon information theory. Based on this quantity, we propose the **Convergence-Aware Entropy-Bounded Sampler**, a training-free sampler for uniform-state dLLMs that prioritizes converged positions while considering both confidence and joint dependency. Across reasoning, coding, and instruction-following benchmarks, existing methods range from a 78.2% increase to a 7.0% reduction in denoising steps, whereas our method consistently reduces them by 9.9%–28.9%. Further, our experiments show that accepting converged positions accelerates denoising by reducing the entropy of neighboring positions, thereby rapidly resolving uncertainty across the entire canvas.

1 INTRODUCTION

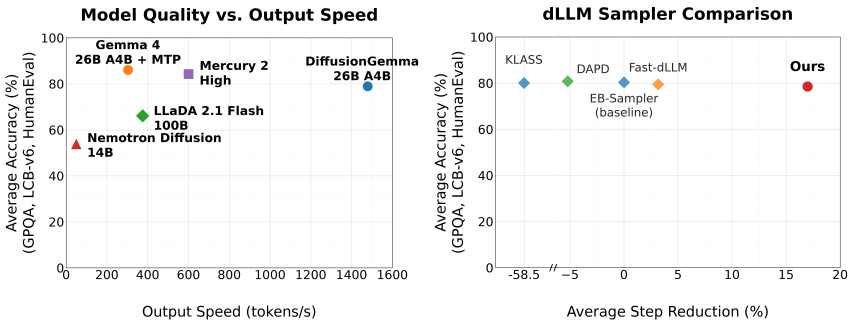
Diffusion LLMs (dLLMs) alleviate the memory bottleneck of autoregressive (AR) LLMs through block-wise parallel generation (Nie et al., 2025; Team et al., 2026a). They generate multiple tokens in parallel over iterative denoising passes; consequently, their speed depends on the number of denoising steps and their canvas size (Team et al., 2026a). As shown in Figure 1a, the early dLLMs were predominantly based on masked dLLMs (Sahoo et al., 2024; Nie et al., 2025; Ou et al., 2025), with models such as LLaDA 2.1 Flash 100B (Bie et al., 2026) and Mercury 2 (Inception Labs, 2026) achieving competitive accuracy-throughput trade-offs against AR models such as Gemma 4 (Team et al., 2026b) equipped with multi-token prediction (MTP) (Gloeckle et al., 2024; Lacombe & Grootendorst, 2026). More recently, DiffusionGemma (Team et al., 2026a), which is based on uniform-state dLLMs (Sahoo et al., 2025; Ye et al., 2026), has demonstrated a competitive accuracy-throughput trade-off across both masked dLLMs and MTP models.

In the denoising process of dLLMs, token acceptance is typically based on single-step statistics (Wu et al., 2026; Ben-Hamu et al., 2025). Thus, positions whose distributions have converged may remain unaccepted if those distributions are not sufficiently sharp. As shown in section 3.1, even a single such position can cause redundant denoising steps. Therefore, a sampling policy that prioritizes convergence over single-step statistics can reduce redundant denoising steps.

However, prior methods (e.g., Fast-dLLM (Wu et al., 2026), EB-Sampler (Ben-Hamu et al., 2025), DAPD (Kim et al., 2026a), and KLASS (Kim et al., 2025)) do not address this issue. This is because existing approaches either ignore probability distributions of consecutive denoising steps or adopt conservative criteria designed for masked dLLMs. As a result, existing methods are limited in their ability to further reduce redundant denoising steps.

In this work, we exploit a re-noising property of uniform-state dLLMs, which enables error correction during the denoising process (Sahoo et al., 2025; Team et al., 2026a), to reduce steps by aggressively accepting positions whose probability distributions have converged. To identify them,

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.



(a) Model accuracy vs. output speed. (b) Sampler performance comparison.

Figure 1: **Trends in dLLMs and performance comparison of dLLM samplers.** (a) The data are reconstructed from the DiffusionGemma Technical Report (Team et al., 2026a). Accuracy is evaluated as the average performance across GPQA Diamond, LiveCodeBench-v6, and HumanEval, and output speed is measured in tokens per second (TPS). (b) Performance comparison of existing dLLM samplers based on the experimental results in section 4.2. The data are computed over the same three benchmarks as in (a).

we propose the **Convergence Probability**, which represents the likelihood of convergence between probability distributions across consecutive steps, through a probabilistic interpretation based on the method of types in Shannon information theory (Csiszar, 1998). This maps KL divergence from an unbounded metric to a bounded probability-domain representation. We further introduce the **Convergence-Aware Entropy-Bounded Sampler**, a training-free sampler for uniform-state dLLMs that incorporates the Convergence Probability into the EB-Sampler (Ben-Hamu et al., 2025) without modifying its original acceptance criterion, while making converged positions more likely to be prioritized under the majority rule. Our approach prioritizes convergence while considering all three sampling criteria: convergence, confidence, and joint dependency.

As shown in Figure 1b, our method reduces the number of denoising steps by 9.9% to 28.9% on DiffusionGemma and consistently outperforms other training-free samplers, including Fast-dLLM, EB-Sampler, KLASS, and DAPD, while maintaining comparable accuracy across all samplers.

2 Backgrounds

2.1 Characteristics of Uniform-state Diffusion LLMs

Uniform-state dLLMs are trained to recover the original token sequence $\mathbf{x}_0$ from a noisy canvas (Team et al., 2026a; Sahoo et al., 2025). Here, $\mathbf{x}_0 \in \mathcal{V}^L$, where $\mathcal{V}$ is the vocabulary and L is the sequence length. During training, a noise level $t \sim \mathrm{Uniform}(0, 1)$ is sampled and each token is independently corrupted with probability t, yielding a noisy canvas $\mathbf{x_t}$. Let, p'_θ denote the previous-step prediction and C_{prev} denote the previous denoised canvases. The loss function is defined as the cross-entropy between the model predictions and the ground-truth canvas obtained from the training data (Team et al., 2026a; Sahoo et al., 2025), as shown in Eq. 1:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{L} \log p_\theta \left(x_0^i \mid \mathbf{x_t}, p'_\theta, C_{\mathrm{prev}} \right). \tag{1}$$

Here, i denotes a token position and p_θ denotes the predicted probability distribution of the model. Then, this objective trains p_θ to approximate the training data distribution p_{data}. This means that the fundamental nature of the denoising process is convergence toward the probability distribution learned by the model. The design philosophy in section 3.3 is based on this fact.

Uniform-state dLLMs allow error correction during inference (Sahoo et al., 2025; Deschenaux et al., 2026). In masked dLLMs with absorbing-state discrete diffusion (Austin et al., 2021; Sahoo et al., 2024; Lou et al., 2024), once a masked position is accepted as a token, the decision is irreversible. In contrast, uniform-state dLLMs allow any accepted position to be re-noised if it later

becomes unstable (Team et al., 2026a). This reversibility of uniform-state dLLMs underlies our acceleration strategy described later in section 3.4.

2.2 Primary Samplers for Masked Diffusion LLMs

A sampler determines which positions are generated in parallel at the current denoising step by its own criterion. The strictness of the acceptance criterion affects both the number of denoising steps and generation quality, creating a trade-off between them. Sampling criteria can be categorized into **confidence**, **joint dependency**, and **convergence**. Confidence measures how certain the model is about its prediction. Joint dependency represents the mismatch between the joint and the marginal distributions across the canvas. Convergence measures the similarity between probability distributions across consecutive denoising steps. Considering any one criterion in isolation can increase uncertainty with respect to the others, potentially requiring additional denoising steps. Representative samplers include Fast-dLLM, EB-Sampler, DAPD, and KLASS, but all of them omit at least one of these criteria, as shown in Table 1.

Table 1: Comparison of three sampling criteria and hyperparameters used by dLLM samplers.

Sampler	Confidence	Joint Dependency	Convergence	# Hyperparams
Fast-dLLM	✓	✗	✗	1
EB-Sampler	✓	✓	✗	1
DAPD	✓	✓	✗	3–4
KLASS	✓	✗	✓	2
Ours	✓	✓	✓	**0 new**

Fast-dLLM is a representative sampler based on single-step confidence (Wu et al., 2026). At each denoising step, a token is accepted if the maximum probability in its predicted distribution exceeds a fixed threshold. Fast-dLLM provides a simple acceptance criterion, but it considers neither joint dependency nor convergence across consecutive steps.

EB-Sampler uses single-step joint dependency as a sampling criterion. EB-Sampler first sorts the masked tokens in the canvas in ascending order of entropy and then accepts the largest set $\mathcal{U}$ that satisfies the entropy bound condition (Ben-Hamu et al., 2025) in Eq. 2.

$$\underbrace{D_{\mathrm{KL}}\left(p_\theta(x^{\mathcal{U}} \mid \mathbf{x}) \,\middle\|\, \prod_{l\in\mathcal{U}} p_\theta(x^l \mid \mathbf{x})\right)}_{\text{joint dependence error}} \le \underbrace{\sum_{l\in\mathcal{U}} H\big(p_\theta(x^l \mid \mathbf{x})\big) - \max_{l\in\mathcal{U}} H\big(p_\theta(x^l \mid \mathbf{x})\big) \le \gamma.}_{\text{entropy bound condition}} \quad (2)$$

This criterion is based on the fact that the joint dependence error in Eq. 2 is precisely equal to the joint mutual information, while the entropy bound condition provides an upper bound on it. Therefore, the entropy bound condition guarantees a set $\mathcal{U}$ whose joint dependence error remains below γ. However, the EB-Sampler does not consider convergence across consecutive steps.

DiffusionGemma further augments the EB-Sampler with **temperature annealing** and an **adaptive stopping** heuristic (Team et al., 2026a). In temperature annealing, the logits are divided by a temperature T, which decreases linearly from $T_{\max}(= 0.8)$ at the first step to $T_{\min}(= 0.4)$ at the maximum step$(= 48)$. This progressively sharpens the probability distributions as denoising proceeds and helps prevent the process from stagnating. Adaptive stopping enables early termination before it reaches the maximum step. It requires the following two conditions to be satisfied simultaneously:

- **Confident predictions:** The mean predictive entropy over the entire canvas falls below a predefined threshold, $e_{\text{stop}} = 0.005$.
- **Stable predictions:** The most likely tokens remain identical across two consecutive denoising steps.

Adaptive stopping considers convergence across consecutive steps through stable predictions, but this criterion is treated as secondary to the entropy bound condition.

DAPD addresses the joint dependency problem by sampling independent sets from a dependency graph constructed using an Attention-Induced Markov Random Field at each step (Kim et al., 2026a). DAPD further formulates the minimum-step denoising problem as a graph-coloring problem and solves it using the Welsh-Powell algorithm. Despite this, DAPD also does not consider convergence across consecutive steps.

KLASS jointly considers confidence and convergence across consecutive denoising steps (Kim et al., 2025). It uses the maximum token probability for confidence and the KL divergence between consecutive-step distributions for convergence. However, since KLASS is designed for masked dLLMs, it employs a more conservative convergence criterion. In addition, it does not consider joint dependency.

Section 3 explains how our method both considers all three criteria shown in Table 1 and accelerates denoising.

3 Convergence-Aware Entropy-Bounded Sampler

3.1 Problems Arising from Not Prioritizing Convergence

This subsection addresses how redundant steps arise when convergence is not prioritized.

During the denoising process of dLLMs, some positions may have converged yet fail the confidence requirement because their distributions remain broad. Figure 2(a) (left) shows how delayed acceptance of a converged position causes redundant steps: position 218 was repeatedly re-accepted after step 11 despite maintaining the same argmax from steps 6 to 8, but was not accepted earlier because other positions had lower entropy. Figure 2(a) (right) shows delayed adaptive stopping caused by an insufficiently peaked probability distribution. Although the canvas had remained unchanged since step 11, the distribution was not sharp enough to satisfy the entropy bound condition. As the temperature decreased over subsequent steps, those positions were accepted, immediately satisfying the adaptive stopping conditions.

Maximum token probability or entropy cannot distinguish an unconverged position from a converged position with a broad distribution. These metrics only indicate how strongly the probability mass is concentrated on a particular token. Additionally, these conditions merely wait for the distribution to become sharper. However, this does not mean that the remaining uncertainty was resolved.

Therefore, convergence across consecutive steps should be prioritized over single-step confidence to reduce the redundant denoising steps caused by the issues described above. Fundamentally, denoising is the process of converging from noise toward the learned model distribution p_{model}, making it natural to consider convergence first.

3.2 Convergence Probability via the Method of Types

This section provides **a more analytical basis for determining the value and granularity of the convergence criterion.** To this end, we transform the KL divergence into a probability domain, thereby **constraining its unbounded range** $[0, \infty)$ **to the bounded range** $(0, 1]$. This allows H_C in section 3.3 to be formulated as an expectation based on P_C. In addition, it allows us to set the criterion as $P_C \geq 0.5$ based on the majority rule, even though P_C could otherwise be tuned through a parameter sweep, thereby providing a guideline that does not need to be retuned across different models and tasks.

In Shannon information theory, the concept of a type is defined as follows (Cover & Thomas, 2006). For a sequence $x^n \in \mathcal{X}^n$ of length n, its empirical distribution is defined as:

Definition 1 (Empirical distribution)**.** *For $x^n \in \mathcal{X}^n$, we write $\hat{P}_{x^n}$ as the empirical distribution of x^n:*

$$\hat{P}_{x^n}(a) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(x_i = a), \qquad \forall a \in \mathcal{X}. \tag{3}$$

Let $\mathcal{P}_n$ be the set of all possible empirical distributions $\hat{P}_{x^n}$ for sequences of length n. Given $P \in \mathcal{P}_n$, the type T_P is defined as the set of all sequences x^n whose empirical distribution is P:

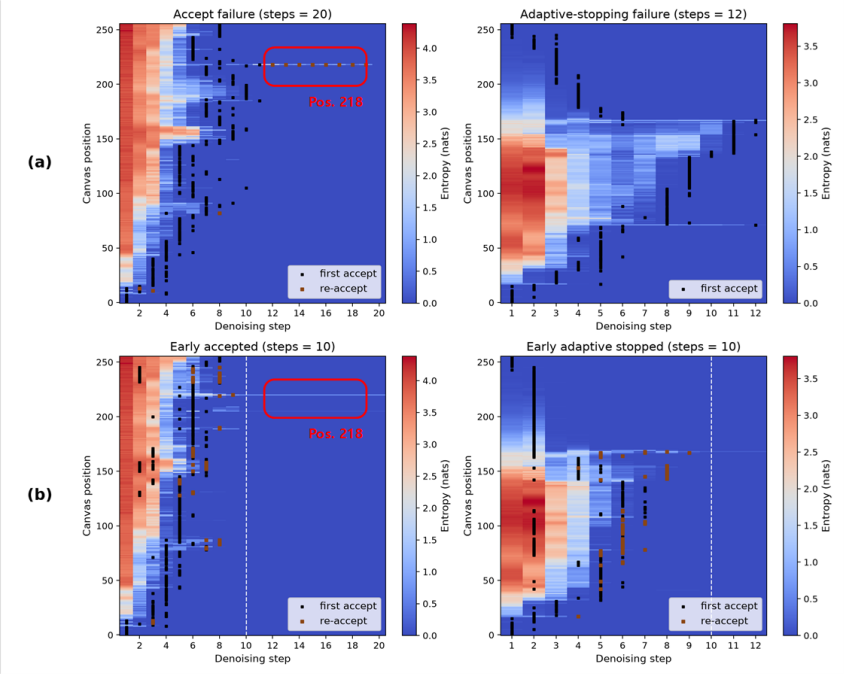


Figure 2: **Entropy heatmaps illustrating redundant denoising steps in DiffusionGemma** (a) Failure of early acceptance at a converged position (left) and failure of early adaptive stopping due to insufficient confidence (right). (b) Problems resolved by our method. Black marks denote first-accepted positions, brown marks denote re-accepted positions, and the white dashed line marks the step at which denoising terminates.

Definition 2 (Type)**.** *For a distribution $P \in \mathcal{P}_n$, the type-P is a subset of $\mathcal{X}^n$, containing all length-n sequences of empirical distribution P:*

$$T_P \triangleq \left\{ x^n \in \mathcal{X}^n : \hat{P}_{x^n} = P \right\}. \tag{4}$$

When $x^n \sim Q^n$, the probability that $\hat{P}_{x^n} = P$, referred to as the probability of a type P, is given by the following lemma (Csiszar, 1998):

Lemma 1. *For all $x^n \in T_P$, the probability of the sequence under the distribution Q^n, the probability of a type $Q^n(T_P)$ is:*

$$Q^n(T_P) \doteq e^{-nD_{KL}(P\|Q)}. \tag{5}$$

We observe that **the token probability distribution** can be considered as **a distributional analogue of the empirical distribution** P. From this perspective, since the token probability distribution is obtained from a single denoising step, it can be considered as the $n = 1$ case. Moreover, since Q corresponds to the learned model distribution p_{model}, the relation for the probability of a type

$$Q(T_P) \approx e^{-D_{\text{KL}}(P \,\|\, Q)}. \tag{6}$$

can be reinterpreted as a measure of the likelihood that the current token distribution has converged toward p_{model}.

In practice, however, the model distribution is unknown, and only the token probability distributions at the current denoising step t, P_t, and the previous step $t-1$, P_{t-1}, are available. Since P_t is obtained at a later stage of the denoising process, we assume that it is closer to the model distribution than P_{t-1}. Accordingly, we approximate $Q \approx P_t$ and $P \approx P_{t-1}$. Under this approximation, the Convergence Probability P_C, defined as the probability that the token probability distributions across consecutive steps have converged, can be written as

$$P_C = P_t\big(T_{P_{t-1}}\big) \approx e^{-D_{\text{KL}}(P_{t-1} \,\|\, P_t)}. \tag{7}$$

3.3 CONVERGENCE-AWARE ENTROPY-BOUNDED SAMPLER

In this section, we introduce the **Convergence-Aware Entropy-Bounded Sampler** using the Convergence Probability P_C. Its design philosophy is as follows:

- **Prioritize convergence.** Exploit converged positions even before their distributions become sharp. This is based on the nature of the denoising process described in section 2.1.
- **Consider all three criteria.** The sampler simultaneously considers convergence, joint dependency, and confidence.

First, we define the **Convergence Entropy** H_C, which converts P_C into an entropy-equivalent quantity, as follows:

- Let $C \sim \text{Bernoulli}(P_C)$ denote the convergence event:

$$C = \begin{cases} \text{converged}, & \text{with probability } P_C, \\ \text{not converged}, & \text{with probability } 1 - P_C. \end{cases}$$

- Define the entropy-equivalent uncertainty $H(C)$ as

$$H(C) = \begin{cases} 0, & C = \text{converged}, \\ H_{\text{th}}, & C = \text{not converged}. \end{cases}$$

- The Convergence Entropy is then defined the an expectation of $H(C)$

$$H_C = \mathbb{E}[H(C)] = P_C \cdot 0 + (1 - P_C)H_{\text{th}} = (1 - P_C)H_{\text{th}}.$$

When $P_C \geq 0.5$, H_C is designed to satisfy the confidence criterion of adaptive stopping (Team et al., 2026a). Therefore, H_{th} is derived from the confidence criterion of adaptive stopping e_{stop} as follows:

$$H_{\text{th}} = \frac{e_{\text{stop}}}{1 - 0.5}. \tag{8}$$

With $e_{\text{stop}} = 0.005$, this yields $H_{\text{th}} = 0.01$, which is one-tenth of $\gamma = 0.1$; therefore, it is much easier to satisfy the entropy bound condition. Consequently, the convergence criterion remains relatively permissive while giving converged positions a higher chance of being accepted, thereby accelerating the denoising process.

Finally, the Final Entropy H_F, which replaces the original entropy H_L in the EB-Sampler and adaptive stopping criteria, is computed as follows:

$$H_F = \min(H_C, H_L). \tag{9}$$

Taking the minimum of H_C and H_L allows the sampler to use the stronger evidence from either convergence or confidence. Positions with different argmax values across consecutive steps are considered unconverged and excluded from the above procedure. Figure 3 illustrates the resulting Convergence-Aware Entropy-Bounded sampling process.

The following equations show where each of the three criteria is incorporated in our sampling process for current canvas $\mathcal{C}$:

- **Prioritizing convergence:**

$$H_{\text{th}} = \frac{e_{\text{stop}}}{1 - 0.5} \ (= 0.1\gamma), \quad \arg\max_{x \in \mathcal{V}} P_{t-1}^l(x) = \arg\max_{x \in \mathcal{V}} P_t^l(x), \ \forall l \in \mathcal{C}. \tag{10}$$

- **Joint dependency:**

$$\sum_{l \in \mathcal{U}} H_F^l - \max_{l \in \mathcal{U}} H_F^l \leq \gamma. \tag{11}$$

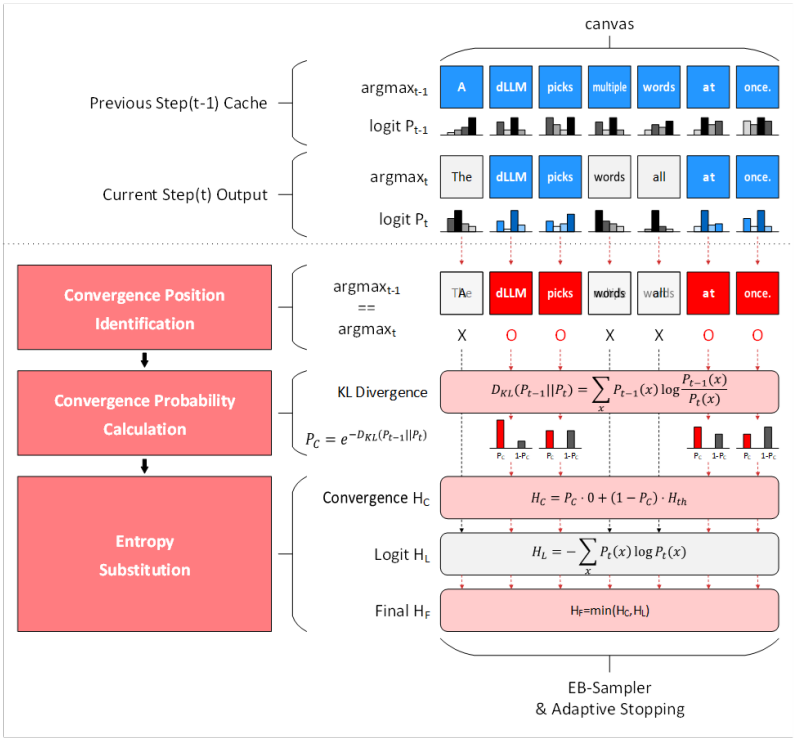


Figure 3: **Overview of Convergence-Aware Entropy-Bounded sampling process**

- **Confidence:**

$$\frac{1}{|\mathcal{C}|} \sum_{l \in \mathcal{C}} H_F^l \leq e_{\text{stop}}. \tag{12}$$

By prioritizing convergence as described above, we eliminated the redundant steps observed in Figure 2(a), as shown in Figure 2(b).

In addition, rather than introducing an arbitrary criterion unrelated to the logit distribution, our method accelerates denoising by allowing accepted converged anchors to sharpen neighboring distributions through bidirectional attention, thereby reducing the entropy of the entire canvas. This effect is validated in section 4.3.

3.4 Synergy with Uniform-State Diffusion LLMs

As discussed in section 2, a majority-rule-based decision is safe only for uniform-state dLLMs. In masked dLLMs, token acceptance is irreversible, so a false positive becomes a permanent error. This is why KLASS requires conservative dual thresholds (Kim et al., 2025). In contrast, uniform-state dLLMs allow accepted tokens to be re-noised, making acceptance reversible and substantially reducing the cost of false positives. This motivates applying our majority-rule strategy specifically to uniform-state dLLMs.

4 Experiments

4.1 Settings

In this work, we evaluate the Convergence-Aware EB-Sampler on DiffusionGemma, an open-weight uniform-state dLLM (Team et al., 2026a), using thinking mode in all experiments.

We compare our method against several training-free baselines, including the native EB-Sampler of DiffusionGemma, Fast-dLLM, KLASS, and DAPD. For each baseline, we use the most favorable

hyperparameter settings reported in the original work or provided in the released implementation. The main hyperparameters are configured as follows:

- **Native EB-Sampler.** We use the default $\gamma = 0.1$ (Team et al., 2026a).
- **Fast-dLLM.** We set the confidence threshold to 0.9 (Wu et al., 2026).
- **KLASS.** The original work reports a confidence threshold sweep from 0.5 to 0.9 and a KL-divergence threshold sweep from 0.001 to 0.01 (Kim et al., 2025). We use the most favorable values for throughput within these ranges, setting the confidence threshold to 0.5 and the KL-divergence threshold to 0.01.
- **DAPD.** We configure the hyperparameters based on the released source code. We set $\texttt{layer_ratio} = 0.3$, meaning that attention scores from the final 30% of decoder layers are used to construct the dependency graph. The edge threshold is dynamically adjusted as
$$\tau = \tau_{\min} + (\tau_{\max} - \tau_{\min}) \cdot \texttt{accepted_ratio}, \tag{13}$$
where $\texttt{accepted_ratio}$ denotes the fraction of positions already accepted in the current canvas. We use $\tau_{\min} = 0.01$ and $\tau_{\max} = 0.15$, following the ParallelBench setting in the released implementation (Kim et al., 2026b). Once 50% of the canvas has been denoised, DAPD switches to the Fast-dLLM sampling strategy to accelerate denoising.

All benchmark evaluations are performed until an EOS token is generated, with the maximum generation length set to 16,384 tokens. We evaluate reasoning capability on MMLU-Pro (Wang et al., 2024), MMMLU (OpenAI, 2024), GPQA Diamond (Rein et al., 2024), and AIME 2026 (Zhang & Math-AI Team, 2026); coding capability on HumanEval (Chen et al., 2021), LiveCodeBench-v6 (Jain et al., 2025), and BigCodeBench (Zhuo et al., 2025); and instruction-following capability on IFEval (Zhou et al., 2023). We evaluate the full benchmark set by default. For MMLU-Pro and MMMLU, due to their large dataset sizes, we sample 500 examples from each benchmark while preserving the original category proportions. We report the average results over three seeds (avg@3). The reported evaluation metrics are the average accuracy, the average number of denoising steps per question, and the average tokens per forward pass (TPF). Each experiment is conducted on either two NVIDIA A6000 48GB GPUs or one NVIDIA A100X 80GB GPU.

4.2 Results

Table 2: Comparison of sampling methods across eight benchmarks. Values in parentheses indicate the relative change from the native EB-Sampler. Higher accuracy and TPF and fewer steps are better.

Benchmark	Metric	EB-Sampler	Fast-dLLM	KLASS	DAPD	**Ours**
MMLU-Pro (n=500)	Acc.(%) ↑	78.4 ± 1.0	77.8 ± 1.7	77.3 ± 0.1	77.4 ± 0.6	75.9 ± 0.8
	Steps ↓	129 (0.0%)	128 (-0.8%)	196 (+51.9%)	132 (+2.3%)	**102 (-20.9%)**
	TPF ↑	19.5 (0.0%)	19.5 (0.0%)	11.3 (-42.1%)	18.0 (-7.7%)	**22.1 (+13.3%)**
MMMLU (n=500)	Acc.(%) ↑	83.7 ± 0.2	83.5 ± 0.8	83.9 ± 0.3	82.5 ± 0.7	83.1 ± 0.4
	Steps ↓	53 (0.0%)	53 (0.0%)	89 (+67.9%)	54 (+1.9%)	**40 (-24.5%)**
	TPF ↑	16.6 (0.0%)	16.7 (+0.6%)	8.4 (-49.4%)	15.6 (-6.0%)	**19.5 (+17.5%)**
GPQA Diamond	Acc.(%) ↑	70.0 ± 1.9	68.0 ± 2.4	68.7 ± 0.8	69.7 ± 0.4	67.8 ± 2.7
	Steps ↓	329 (0.0%)	310 (-5.8%)	464 (+41.0%)	329 (0.0%)	**234 (-28.9%)**
	TPF ↑	18.3 (0.0%)	18.7 (+2.2%)	11.8 (-35.5%)	17.0 (-7.1%)	**20.6 (+12.6%)**
AIME 2026	Acc.(%) ↑	70.0 ± 0.0	68.9 ± 9.6	66.7 ± 7.2	64.4 ± 5.7	70.0 ± 2.7
	Steps ↓	402 (0.0%)	374 (-7.0%)	625 (+55.5%)	478 (+18.9%)	**354 (-11.9%)**
	TPF ↑	21.2 (0.0%)	22.2 (+4.7%)	14.0 (-34.0%)	18.3 (-13.7%)	**22.3 (+5.2%)**
HumanEval	Acc.(%) ↑	94.3 ± 0.8	94.7 ± 0.8	94.9 ± 0.8	95.7 ± 2.6	93.9 ± 0.9
	Steps ↓	58 (0.0%)	55 (-5.2%)	96 (+65.5%)	64 (+10.3%)	**52 (-10.3%)**
	TPF ↑	26.0 (0.0%)	26.9 (+3.5%)	14.7 (-43.5%)	23.6 (-9.2%)	**28.0 (+7.7%)**
LiveCodeBench-v6	Acc.(%) ↑	76.8 ± 0.8	75.8 ± 1.1	76.6 ± 0.0	77.1 ± 0.8	73.9 ± 1.2
	Steps ↓	280 (0.0%)	264 (-5.7%)	407 (+45.4%)	291 (+3.9%)	**248 (-11.4%)**
	TPF ↑	22.5 (0.0%)	23.0 (+2.2%)	13.8 (-38.7%)	20.7 (-8.0%)	**24.3 (+8.0%)**
BigCodeBench	Acc.(%) ↑	46.4 ± 0.7	44.9 ± 0.3	46.7 ± 0.4	46.6 ± 0.3	45.4 ± 0.6
	Steps ↓	91 (0.0%)	89 (-2.2%)	148 (+62.6%)	96 (+5.5%)	**82 (-9.9%)**
	TPF ↑	21.4 (0.0%)	22.0 (+2.8%)	12.6 (-41.1%)	20.0 (-6.5%)	**23.5 (+9.8%)**
IFEval (inst-strict)	Acc.(%) ↑	95.1 ± 0.4	95.0 ± 0.1	95.2 ± 0.3	95.5 ± 0.3	94.1 ± 0.3
	Steps ↓	78 (0.0%)	79 (+1.3%)	139 (+78.2%)	77 (-1.3%)	**64 (-17.9%)**
	TPF ↑	18.5 (0.0%)	18.7 (+1.1%)	9.8 (-47.0%)	18.0 (-2.7%)	**21.5 (+16.2%)**

As shown in Table 2, our **Convergence-Aware EB-Sampler achieves the largest reduction in denoising steps and the highest TPF across all baselines.** The number of denoising steps for

the existing baselines ranges from a 78.2% increase in the worst case to only a 7.0% reduction in the best case compared with the native EB-Sampler. In contrast, the Convergence-Aware EB-Sampler achieves at least a 9.9% step reduction on BigCodeBench, already exceeding the best reduction achieved by the other baselines, and reaches a maximum reduction of 28.9% on GPQA Diamond. Considering the variation across seeds, accuracy remains similar across the baselines and our method.

4.3 Discussions

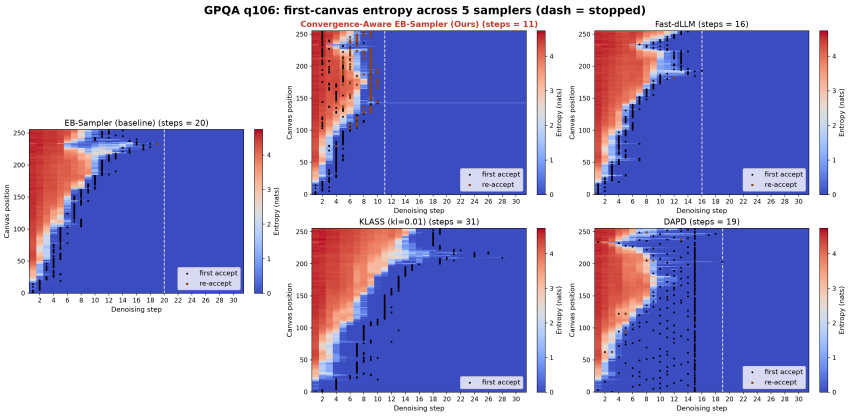


Figure 4: **Entropy heatmaps of the Convergence-Aware EB-Sampler and baselines.** The results are shown for the first canvas for Question 106 of the GPQA Diamond benchmark. Black marks denote first-accepted positions, brown marks denote re-accepted positions, and the white dashed line marks the step at which denoising terminates.

Figure 4 presents entropy heatmaps for the baselines and the Convergence-Aware EB-Sampler, illustrating the relationship between each sampling strategy and the point at which uncertainty in the canvas is resolved. For both the Convergence-Aware EB-Sampler and Fast-dLLM, final acceptance generally aligns with the point at which uncertainty is resolved according to the entropy value. In the Convergence-Aware EB-Sampler, however, some converged positions are accepted earlier as anchors even before their uncertainty is fully resolved. These anchors rapidly reduce the entropy of neighbors, thereby reducing the number of steps required to resolve uncertainty. In contrast, Fast-dLLM does not provide such intermediate anchors and therefore does not rapidly reduce the canvas entropy. KLASS and DAPD show final acceptance patterns that are less aligned with the point at which uncertainty is resolved in the entropy distribution. KLASS tends to accept positions after their uncertainty has already been resolved, indicating remaining room for further acceleration. DAPD samples positions more uniformly across the canvas because its decisions are based on a dependency graph. However, because its sampling criterion is not directly aligned with entropy-based uncertainty, denoising also tends to terminate after the uncertainty has already been resolved.

5 Conclusions

We presented the Convergence-Aware Entropy-Bounded Sampler, the first uniform-state dLLM sampler to prioritize convergence while jointly considering confidence and joint dependency. To this end, we transformed the KL divergence into a Convergence Probability through a probabilistic interpretation based on the method of types in Shannon information theory. This transformation converts an unbounded metric into a bounded probability-domain quantity and provides a more analytical basis for determining the criterion value and granularity. We then integrated it into the EB-Sampler with a convergence-prioritized design while preserving the original conditions, thereby incorporating both joint dependency and confidence criteria. Finally, our experiments demonstrated that our method achieves the largest reduction in denoising steps among the evaluated samplers on the uniform-state dLLM. We further observe that accepted converged positions reduce the entropy of neighboring positions, thereby rapidly resolving uncertainty across the entire canvas.

AI USE STATEMENT

All research ideas, mathematical reasoning and validation, sampler design, experimental design, result analysis, and figure preparation were conducted by the authors. AI tools were used to assist with code generation and experiment execution; however, all generated code and experimental scripts were reviewed, modified where necessary, and executed by the authors. **The paper outline and drafts were written in the authors' native language** and subsequently translated into English with the assistance of AI tools. AI tools were also used to refine the wording and improve the clarity of the manuscript. References were retrieved with the assistance of AI and reviewed by the authors. The reproducibility statement was drafted by AI and reviewed by the authors.

REPRODUCIBILITY STATEMENT

For MMLU-Pro and MMMLU, we evaluate on a fixed subset of 500 questions from each benchmark rather than the full test set. We select these subsets by stratified sampling with a fixed random seed (0), so the selection is deterministic and can be reproduced exactly. MMLU-Pro is stratified by subject category and MMMLU by language: from each group we take a number of questions proportional to that group's size in the full set (rounded, with at least one question per group), and then keep 500 in total. This preserves the category/language mix of the original benchmark. We use the test split of MMLU-Pro, and the test files of MMMLU. Each question is formatted as the question followed by its options (A, B, ...), and the gold answer is the correct option letter. We use random seeds by default for avg@3. Before evaluating the samplers, then we check whether the baseline results are comparable to those reported in the DiffusionGemma Technical Report. Therefore, for these two benchmarks, we select three seeds whose accuracies are closest to the reported results. For MMLU-Pro, MMMLU, and GPQA Diamond (multiple-choice) and AIME (integer answer), we grade by extracting the model's final answer and comparing it to the gold answer by exact match. We first remove the model's thinking and parse only the final answer. For the multiple-choice benchmarks we take the answer letter — preferring an explicit "The answer is (X)" statement, and otherwise falling back to the last standalone option letter in the response — and count the question correct if this letter matches the gold option (A–J for MMLU-Pro, A–D for MMMLU and GPQA Diamond). For AIME we take the final integer (from "The answer is N" or '$\boxed{N}$', otherwise the last integer in the response) and count it correct if it equals the gold integer. Accuracy is the fraction of correct questions. The code benchmarks (HumanEval, LiveCodeBench-v6, BigCodeBench) and IFEval are graded with their official evaluators.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas S Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 55981–56007. Curran Associates, Inc., 2025. doi: 10.52202/085713-1874. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/510e22f4a2da5212a64f1591736e2eaf-Paper-Conference.pdf.

Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, et al. LLaDA2.1: Speeding up text diffusion via token editing. *arXiv preprint arXiv:2602.08676*, 2026.

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.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2 edition, 2006. doi: 10.1002/047174882X.

I. Csiszar. The method of types [information theory]. *IEEE Transactions on Information Theory*, 44 (6):2505–2523, 1998. doi: 10.1109/18.720546.

Justin Deschenaux, Caglar Gulcehre, and Subham Sekhar Sahoo. The diffusion duality, chapter II: ψ-samplers and efficient curriculum. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RSIoYWIzaP.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Roziere, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15706–15734. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/gloeckle24a.html.

Inception Labs. Introducing mercury 2. https://www.inceptionlabs.ai/blog/introducing-mercury-2, February 2026. Accessed: 2026-09-24.

Naman Jain, 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. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/94074dd5a072d28ff75a76dabed43767-Paper-Conference.pdf.

Bumjun Kim, Dongjae Jeon, Moongyu Jeon, and Albert No. DAPD: Dependency-Aware Parallel Decoding via Attention for Diffusion LLMs. In *International Conference on Machine Learning*, 2026a.

Bumjun Kim, Dongjae Jeon, Moongyu Jeon, and Albert No. DAPD: Dependency-Aware Parallel Decoding via Attention for Diffusion LLMs. https://github.com/quasar529/DAPD, 2026b. GitHub repository, accessed September 25, 2026.

Seo Hyun Kim, Sunwoo Hong, Hojung Jung, Youngrok Park, and Se-Young Yun. KLASS: KL-Guided Fast Inference in Masked Diffusion Models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 92267–92301. Curran Associates, Inc., 2025. doi: 10.52202/085713-3087. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/8514c1f12dcdd061ac8521feb78f28ca-Paper-Conference.pdf.

Olivier Lacombe and Maarten Grootendorst. Accelerating gemma 4: Faster inference with multi-token prediction drafters. https://blog.google/innovation-and-ai/technology/developers-tools/multi-token-prediction-gemma-4/, May 2026. Google Blog, accessed September 21, 2026.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/lou24a.html.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan LI. Large language diffusion models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 50608–50646. Curran Associates, Inc., 2025. doi: 10.52202/085713-1689. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/48b383b24230e0e6e649d9c98dae4d8c-Paper-Conference.pdf.

OpenAI. MMMLU: Multilingual Massive Multitask Language Understanding. https://huggingface.co/datasets/openai/MMMLU, 2024. Hugging Face dataset, accessed September 26, 2026.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations*, volume 2025, pp. 64972–65009, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T Chiu, and Volodymyr Kuleshov. The diffusion duality. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 52584–52619. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/sahoo25a.html.

DiffusionGemma Team, Adrien Ali Taïga, James Assiene, Daniele Calandriello, Rahma Chaabouni, João Gante, Tamara von Glehn, Nate Keating, Chris Knutsen, Martin Kukla, et al. DiffusionGemma Technical Report. *arXiv preprint arXiv:2608.00146*, 2026a.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026b.

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. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290. Curran Associates, Inc., 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad236edc564f3e3156e1b2feafb99a24-Paper-Datasets_and_Benchmarks_Track.pdf.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dLLM: Training-Free Acceleration of Diffusion LLM by Enabling KV Cache and Parallel Decoding. In *International Conference on Learning Representations*, volume 2026, pp. 57027–57051, 2026.

Mengyu Ye, Keito Kudo, Wataru Ikeda, Ryosuke Matsuda, Keisuke Sakaguchi, and Jun Suzuki. Sumi: Open uniform diffusion language model from scratch. *arXiv preprint arXiv:2606.19005*, 2026.

Yifan Zhang and Math-AI Team. American Invitational Mathematics Examination (AIME) 2026. https://huggingface.co/datasets/math-ai/aime26, 2026.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

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, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro Von Werra. BigCodeBench: Benchmarking Code Generation with Diverse Function Calls and Complex Instructions. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 66602–66656, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a6a90bcc2aa470c3871b2d39a67d26e8-Paper-Conference.pdf.