DECOUPLE TO DECODE: PARALLEL DECODING WITH DEFERRED PROPOSAL VERIFICATION FOR DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete diffusion language models (dLLMs) alleviate the sequential bottleneck of autoregressive generation by predicting multiple masked tokens in parallel at each denoising step. However, existing parallel decoding methods typically make single-step commitment decisions, coupling whether a prediction is reliable enough to commit with whether it is worth retaining for further evidence, thereby tying broader proposal coverage to looser commitment criteria. Indeed, we observe two complementary regimes: one where predictions highly agree with the final tokens and are suitable for immediate commitment, and another where candidates of varying reliability remain mixed but become more separable under cross-step re-evaluation. Motivated by this observation, we propose **Decouple to Decode (D2D)**, a training-free framework that decouples immediate commitment from deferred verification. D2D uses conditional risk budgets to commit a **Core** and retain a masked **Shell** for one-step verification; in the next ordinary forward pass, Shell proposals are accepted if their token identities remain unchanged and confidence does not decrease, requiring no extra verifier or model evaluation. Experiments on LLaDA and Dream across five benchmarks spanning mathematical reasoning, code generation, and instruction following demonstrate up to $9.92\times$ wall-clock speedup over standard decoding and 64.4% over confidence decoding, while maintaining comparable task performance. Combined with KV caching, D2D achieves up to $16.37\times$ speedup over standard decoding.

1 INTRODUCTION

Discrete diffusion large language models (dLLMs) have emerged as a promising alternative to autoregressive language models by allowing multiple tokens to be predicted and generated in parallel (Lee et al., 2018; Ghazvininejad et al., 2019). At each decoding step, dLLMs predict tokens for all masked positions and reveal a subset of them, progressively completing the sequence (Nie et al., 2026). Committing more tokens per step can substantially reduce the number of decoding steps, but overly aggressive parallel decoding may overlook dependencies among simultaneously predicted positions and consequently degrade generation quality (Wu et al., 2026; Zhang et al., 2026). Efficient dLLM decoding therefore fundamentally depends on identifying which token predictions are reliable enough to be committed as early as possible.

To address this challenge, existing parallel decoding strategies use prediction reliability signals to decide which tokens can be safely committed at each step. A common line of work relies on current-step evidence, such as token-level confidence, margin-based, or entropy-based uncertainty(Wu et al., 2026; Li et al., 2025; Ben-Hamu et al., 2026). More recent methods further exploit temporal signals across decoding steps, including distributional consistency, accumulated prediction histories, and token-identity persistence (Kim et al., 2026; Chen et al., 2025; Wang et al., 2026; Chen et al., 2026). Despite this richer evidence, these signals are still largely used to answer the same immediate question of whether to commit a prediction at the current step. Predictions that fail the current commitment criterion are not retained as explicit proposals but are instead re-predicted at later steps, so exploiting more current predictions requires a looser commitment rule and greater error risk. This raises a natural question: can such predictions be retained and re-evaluated under updated context, allowing reliable candidates to be committed earlier without loosening the current-step criterion?

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.

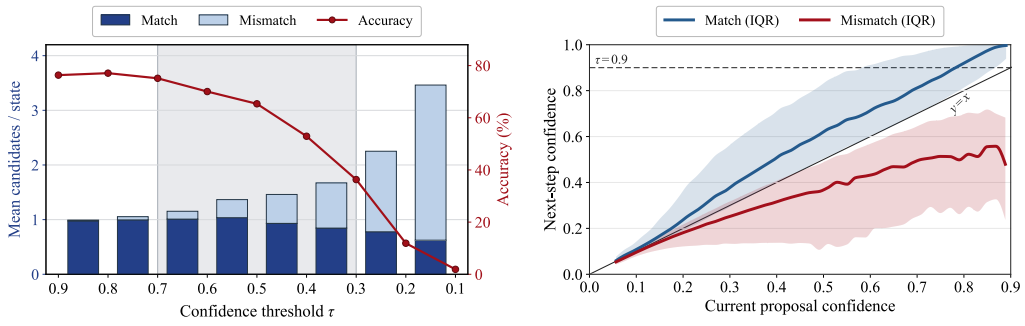

Figure 1: **Current-step confidence and one-step re-evaluation provide complementary evidence.** On LLaDA GSM8K, (a) groups masked positions from confidence decoding trajectories ($\tau = 0.9$) by current confidence, showing their match/mismatch composition and generation accuracy under different thresholds. (b) tracks uncommitted proposals one step forward, comparing current confidence with next-step probability of the same token, grouped by final-token match. Relative to $y = x$, matching proposals tend to gain stronger support than mismatching ones. Lines and shaded regions denote the median and interquartile range (IQR).

To assess this possibility, we analyze confidence-decoding trajectories and track how intermediate predictions evolve across steps. As shown in Figure 1(a), we define a prediction as a match if it agrees with the final decoded token at the same position. High-confidence predictions are dominated by matches, while the intermediate-confidence region contains many matches mixed with mismatches. Lowering the threshold commits more matches earlier, but also admits more mismatches and eventually degrades task performance, indicating that useful proposals remain beyond the immediate commitment region but cannot be reliably separated by current-step confidence alone. Figure 1(b) shows that one-step re-evaluation provides this additional separation: under similar current support, matching proposals tend to receive stronger next-step support than mismatching ones. Together, these observations motivate decoupling immediate commitment from subsequent proposal reassessment, allowing within-step and cross-step evidence to serve distinct roles during decoding.

Building on these insights, we propose **Decouple to Decode (D2D)**, a training-free decoding framework that explicitly separates immediate commitment from subsequent proposal verification. D2D uses conditional risk budgets to organize candidate predictions into a *Core* for immediate commitment, a *Shell* containing proposals retained for reassessment, and a *Tail* that continues through ordinary decoding. After committed tokens update the context, proposals in the previous Shell are verified in the next ordinary forward pass and accepted when their token identities remain unchanged and their confidence does not decrease. The same forward pass also selects a fresh Core and Shell, requiring neither a dedicated verifier nor additional model evaluations. This separation allows D2D to exploit reliable proposals beyond the immediate commitment boundary while controlling the risk of aggressive parallel decoding. Overall, our contributions are summarized as follows:

- We reveal a complementary relationship between within-step and cross-step evidence in dLLM decoding: within-step evidence supports immediate commitment for sufficiently reliable predictions, while cross-step re-evaluation further distinguishes reliable candidates in the mixed region that remains ambiguous at the current step.
- We propose **Decouple to Decode (D2D)**, a training-free framework that decouples immediate commitment from deferred verification through Core–Shell selection. We formulate this process under a shared risk-constrained objective, where the Core is selected under the original commitment budget and the Shell captures additional candidates admitted by a controlled relaxation for subsequent verification.
- We evaluate D2D on two representative dLLMs across five benchmarks spanning mathematical reasoning, code generation, and instruction following. D2D achieves the highest TPS across all settings while matching or exceeding standard decoding in task performance, with up to $9.92\times$ wall-clock speedup over standard decoding and 64.4% over confidence decoding. With KV caching mechanisms, the speedup rises to $16.37\times$.

2 Related Work

Efficient Decoding for dLLMs. Discrete diffusion models generate categorical sequences through iterative denoising, with absorbing-state formulations forming the basis of masked diffusion language modeling (Austin et al., 2021a; Sahoo et al., 2024). Recent models such as LLaDA and Dream scale this paradigm to large language models, enabling parallel prediction over masked positions (Nie et al., 2026; Ye et al., 2025). Existing inference acceleration methods primarily improve either computational reuse or decoding progress per forward pass. Representative approaches to computational reuse include Fast-dLLM and dKV-Cache, which develop diffusion-specific caching mechanisms to reduce redundant computation across decoding steps (Wu et al., 2026; Ma et al., 2026). Adaptive decoding strategies, meanwhile, employ criteria such as token confidence, entropy-based uncertainty, and confidence profiles to increase parallel commitment (Wu et al., 2026; Ben-Hamu et al., 2026; Kasa et al., 2026). Together, these approaches improve decoding efficiency by reducing per-step computation or safely increasing the number of tokens committed in parallel.

Temporal Information in dLLM Decoding. Recent studies show that temporal information can complement within-step signals by revealing how predictions evolve across decoding steps. KLASS (Kim et al., 2026) combines inter-step distributional consistency with current confidence, while CreditDecoding (Wang et al., 2026) accumulates historical evidence into current logits and STDec (Chen et al., 2026) exploits persistent token identities to adapt commitment thresholds. LESS (Mohamed et al., 2026) further integrates confidence, token persistence, and inter-step distributional stability into a joint unmasking rule. These methods primarily use prediction history as auxiliary evidence for the current commitment decision. DCD (Shu et al., 2026) instead defers uncertain positions through a certainty-aware sliding window, allowing them to be re-predicted with additional context. A related draft-and-revise method, WINO (Hong et al., 2026), aggressively unmasks draft tokens and later re-masks suspicious ones. In contrast, D2D retains current predictions as explicit proposals while keeping their positions masked, and verifies the same assignments in the next ordinary forward pass without model retraining or extra evaluations.

3 Method

3.1 Preliminaries

dLLM Formulation. Discrete diffusion language models (dLLMs) generate text by iteratively predicting and revealing tokens in partially masked sequences (Austin et al., 2021a; Sahoo et al., 2024; Nie et al., 2026). Starting from a fully masked response, this process progressively completes the sequence over multiple decoding iterations. Let $\mathcal{V}$ denote the vocabulary containing a special mask token m. Given a prompt P and response length L, let $\mathbf{y}_t = (y_{t,1}, \ldots, y_{t,L}) \in \mathcal{V}^L$ denote the response at iteration t, with $\mathbf{y}_0 = m^L$, and let $M_t = \{i : y_{t,i} = m\}$ denote its masked positions. With bidirectional attention, a single forward pass of the dLLM p_θ predicts all positions in M_t:

$$p_{t,i}(v) = p_\theta^i(v \mid \mathbf{y}_t, P), \qquad \hat{x}_{t,i} = \arg\max_{v \in \mathcal{V}} p_{t,i}(v), \qquad c_{t,i} = p_{t,i}(\hat{x}_{t,i}), \tag{1}$$

where $p_{t,i}$ denotes the predictive distribution at position i, and $\hat{x}_{t,i}$ and $c_{t,i}$ denote the corresponding greedy proposal and confidence, respectively.

Parallel Decoding. A parallel decoding policy converts these predictions into state updates by selecting a commitment set $S_t \subseteq M_t$:

$$y_{t+1}^i = \begin{cases} \hat{x}_{t,i}, & i \in S_t, \\ y_t^i, & i \notin S_t, \end{cases} \qquad M_{t+1} = M_t \setminus S_t. \tag{2}$$

Existing decoding policies mainly differ in how they construct S_t from predictive statistics such as confidence or entropy. All proposals in S_t are generated from the same state $\mathbf{y}_t$ and become visible only in the next iteration, while positions outside S_t remain masked and are re-predicted under the updated context. Thus, a larger S_t increases per-iteration progress but commits more predictions before they can benefit from the updated context, potentially increasing commitment errors.

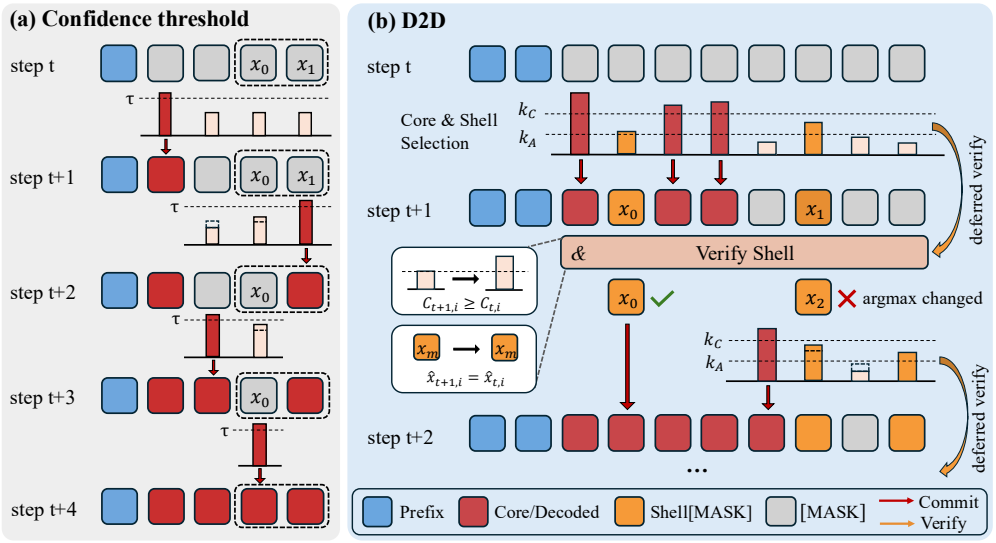

Figure 2: **Overview of D2D decoding framework.** (a) confidence-threshold τ decoding may delay stable proposals until their confidence exceeds τ, which creates a commitment gap. (b) an example of the complete D2D decoding process. Core candidates are committed immediately, while Shell candidates remain masked and are verified by the next step without additional forwards.

3.2 Decouple to Decode

Parallel decoding in Eq. (2) makes a one-step commit-or-repredict decision: selected predictions are exposed to the next state, while the rest remain masked and recomputed. D2D decouples these outcomes by committing a reliable *Core* immediately while retaining a surrounding *Shell* as masked proposals for one-step verification. As illustrated in Figure 2, this allows D2D to reuse predictions beyond the immediate commitment boundary without exposing them before verification. The complete decoding procedure is summarized in Algorithm 1 in Appendix A.

Core–Shell selection. At iteration t, let $U_t \subseteq M_t$ contain the active masked positions and $V_t \subseteq U_t$ those whose proposals from H_{t-1} pass verification. We rank the remaining candidates $E_t = U_t \setminus V_t$ by decreasing confidence and denote their top-k prefix by $E_t^{(k)}$, with $E_t^{(0)} = \varnothing$ and deterministic tie-breaking for both ranking and argmax. Using the aggregate risk score $F_t(S) = 1 - \prod_{i \in S} c_{t,i}$, with $F_t(\varnothing) = 0$, D2D constructs the Core $C_t = E_t^{(k_C)}$ and Shell $H_t = E_t^{(k_A)} \setminus C_t$ through

$$k_C = \max\{k : F_t(V_t \cup E_t^{(k)}) \le \delta\}, \tag{3}$$

$$k_A = \max\{k \ge k_C : F_t(E_t^{(k)} \setminus C_t) \le \epsilon_A[1 - F_t(V_t) - F_t(C_t)]\}, \tag{4}$$

where $k \in \{0, \dots, |E_t|\}$ and $\delta, \epsilon_A \in (0, 1)$ control commitment and Shell admission. By construction, $F_t(V_t) \le \delta$. Shell admission is enabled only when $1 - F_t(V_t) - F_t(C_t) > 0$; otherwise, $H_t = \varnothing$. If $V_t \cup C_t = \varnothing$, no nonempty commitment satisfies the original budget, so we commit the highest-confidence remaining prediction as a progress fallback and set $H_t = \varnothing$. This fallback lies outside the δ-level guarantee but is the minimum-risk nonempty choice (Appendix B.1). Otherwise, D2D commits $S_t = V_t \cup C_t$, while keeping H_t masked for next-step reassessment and leaving candidates beyond k_A in the Tail.

One-step deferred verification. For each $i \in H_t$, D2D stores $(i, \hat{x}_{t,i}, c_{t,i})$ without exposing the proposal. Conditioned on the updated context, the next ordinary forward pass first identifies

$$\widetilde{V}_{t+1} = \{i \in H_t : \hat{x}_{t+1,i} = \hat{x}_{t,i},\ p_\theta^i(\hat{x}_{t,i} \mid y_{t+1}, P) \ge c_{t,i}\}. \tag{5}$$

We rank $\widetilde{V}_{t+1}$ by decreasing updated confidence and retain its largest prefix V_{t+1} satisfying $F_{t+1}(V_{t+1}) \le \delta$. Proposals removed by this budget truncation return to candidate selection under the updated predictions. Thus, every retained verified set satisfies the original risk budget by construction. Verification, budget truncation, and fresh selection reuse the same forward pass, requiring no additional model evaluation.

3.3 Risk-Constrained Selection and Deferred Verification

We analyze Core–Shell selection and verification through a common risk-constrained objective. Fix iteration t, let $X_i^\star$ be the reference token specified independently of D2D's acceptance decisions, and define $q_{t,i} = \Pr(X_i^\star = \hat{x}_{t,i} \mid \mathcal{D}_t)$ given the selection evidence $\mathcal{D}_t$. With V_t fixed, consider

$$\begin{aligned} \max_{S \subseteq E_t} \quad & \mathbb{E}\left[\sum_{i \in S} \mathbf{1}\{X_i^\star = \hat{x}_{t,i}\} \,\middle|\, \mathcal{D}_t \right] \\ \text{s.t.} \quad & \Pr(\exists i \in V_t \cup S : X_i^\star \neq \hat{x}_{t,i} \,|\, \mathcal{D}_t) \leq \rho. \end{aligned} \tag{6}$$

Under $q_{t,i} = c_{t,i}$ and conditional joint independence of the agreement indicators given $\mathcal{D}_t$, the objective becomes $\sum_{i \in S} c_{t,i}$ and the constraint becomes $\sum_{i \in V_t \cup S} -\log c_{t,i} \leq -\log(1-\rho)$.

Optimal selection under confidence bounds. Higher confidence increases reward and lowers risk cost, so replacing a selected candidate with a higher-confidence one preserves feasibility without reducing the objective, yielding prefix optimality (Appendix B.1).

Theorem 3.1 (Prefix optimality)**.** *Under the stated assumptions, for $F_t(V_t) \leq \rho < 1$ and $c_{t,i} \in (0,1]$ for $i \in U_t$, the largest feasible confidence-ranked prefix solves Eq. (6) and has maximum feasible cardinality. With fixed tie-breaking, these prefix solutions are nested as ρ increases.*

The Core C_t is the optimal solution at $\rho = \delta$. Let $A_t = C_t \cup H_t$ and let b_t denote the Shell allowance in Eq. (4), i.e., $b_t = \epsilon_A \max\{1 - F_t(V_t) - F_t(C_t), 0\}$, taken as zero when admission is disabled. Define the induced relaxation as follows:

$$\epsilon_t = \frac{\max\{\delta,\ F_t(S_t) + [1 - F_t(S_t)]b_t\}}{\delta} - 1 \geq 0. \tag{7}$$

For disjoint S_t, H, the identity $F_t(S_t \cup H) = F_t(S_t) + [1 - F_t(S_t)]F_t(H)$ converts Shell admission into a joint-risk bound. Together with Core maximality, this makes A_t the largest feasible prefix at $\rho = (1+\epsilon_t)\delta < 1$ (Appendix B.2), satisfying

$$\prod_{i \in V_t \cup A_t} c_{t,i} \geq 1 - (1+\epsilon_t)\delta. \tag{8}$$

Under the stated probability assumptions, Eq. (8) is exactly the feasibility condition of the optimization problem (6) obtained by relaxing its risk constraint from $\Pr(\exists i \in V_t \cup S : X_i^\star \neq \hat{x}_{t,i} \mid \mathcal{D}_t) \leq \delta$ to $\Pr(\exists i \in V_t \cup S : X_i^\star \neq \hat{x}_{t,i} \mid \mathcal{D}_t) \leq (1+\epsilon_t)\delta$, with the objective and candidate pool unchanged. By Theorem 3.1, C_t and A_t are optimal under the original and relaxed constraints, respectively. Here, ϵ_t is induced by the Shell admission rule, not an additional hyperparameter. Thus, $H_t = A_t \setminus C_t$ contains the additional candidates enabled by this relaxation; since joint commitment with S_t is not certified under the original constraint, D2D retains them for next-step verification.

Posterior-supported verification. Next-step evidence Z_{t+1} updates the posterior of each stored assignment to $q_{t,i}^+ = \Pr(X_i^\star = \hat{x}_{t,i} \mid \mathcal{D}_t, Z_{t+1})$. Under the additional identification $q_{t,i}^+ = p_\theta^i(\hat{x}_{t,i} \mid y_{t+1}, P)$, raw verification retains unchanged greedy assignments with $q_{t,i}^+ \geq q_{t,i}$, after which budget truncation selects a subset satisfying the original risk budget.

Proposition 3.2 (Reward–risk preservation)**.** *For every retained subset $R \subseteq V_{t+1}$,*

$$\begin{gathered} \sum_{i \in R} c_{t+1,i} \geq \sum_{i \in R} c_{t,i}, \qquad \sum_{i \in R} -\log c_{t+1,i} \leq \sum_{i \in R} -\log c_{t,i}, \\ F_{t+1}(V_{t+1}) \leq \min\{F_t(H_t), \delta\} \leq b_t. \end{gathered} \tag{9}$$

These inequalities follow from non-decreasing support and $V_{t+1} \subseteq \widetilde{V}_{t+1} \subseteq H_t$, without requiring the probability assumptions. Under the posterior interpretation, verification therefore preserves or improves the reward–risk terms of Eq. (6) for retained assignments. Moreover, $V_t \cup C_t \cup V_{t+1}$ inherits the admission-time relaxed bound $(1+\epsilon_t)\delta$, since it retains only assignments from $V_t \cup A_t$. Budget truncation further guarantees $F_{t+1}(V_{t+1}) \leq \delta$, so subsequent Core selection always starts from a feasible verified set. Since retained proposals have no higher log-risk cost than at admission, their updated scores cannot reduce the budget available for fresh Core selection, with fresh-candidate scores fixed. Details are given in Appendix B.3.

3.4 DISCUSSION

The analysis above highlights that D2D is distinguished not by whether cross-step information is used, but by how it enters the decoding decision. Existing history-aware methods (Kim et al., 2026; Chen et al., 2026; Mohamed et al., 2026) generally incorporate temporal evidence into the current commitment rule, without a separate mechanism for retaining uncertain predictions for further evaluation. Consequently, exploiting a broader range of candidates is tied to making the immediate commitment rule more permissive. D2D instead decouples these roles through Core–Shell selection: the original risk budget governs immediate commitment, while the relaxed risk region determines which additional proposals are retained for next-step verification. This broadens proposal coverage without enlarging the immediate commitment set, while verification reassesses the same stored assignments under updated context without additional model evaluations. D2D also differs from revocable decoding such as WINO (Hong et al., 2026): tentative drafts are exposed before possible re-masking and may influence other predictions before revocation, whereas D2D keeps Shell proposals masked until verification, preventing unverified proposals from affecting subsequent decoding.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and benchmarks. We evaluate D2D on two representative masked diffusion language models: LLaDA-8B-Instruct(Nie et al., 2026) (hereafter LLaDA), which is pretrained from scratch, and Dream-v0-Instruct-7B(Ye et al., 2025) (hereafter Dream), which is adapted from autoregressive models. Our evaluation covers five benchmarks across three task categories: GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024) for mathematical reasoning; HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021b) for code generation; and IFEval(Zhou et al., 2023) for instruction following. We use the full evaluation set of each benchmark with few-shot prompts and generate a single response per instance.

Baselines. We compare our D2D with five decoding baselines: **Standard** decoding with one token per step, and four parallel decoding strategies including **Confidence**(Wu et al., 2026), **KLASS** (Kim et al., 2026), **WINO** (Hong et al., 2026), and **DCD** (Shu et al., 2026). For each baseline, we adopt the hyperparameters specified in its original implementation or recommended evaluation settings.

Implementation details. We conduct all evaluations using lm-eval-harness. All methods use greedy decoding with a maximum generation length of 256 tokens, and a block size of 32. For D2D, we select the commitment budget δ from $\{0.3, 0.4, 0.5\}$ and the Shell admission parameter ϵ_A from $\{0.7, 0.8, 0.9\}$; for MBPP, we additionally consider $\delta = 0.45$ and $\epsilon_A \in \{0.4, 0.5\}$. Detailed settings for each model and benchmark are provided in Appendix C. We report accuracy on GSM8K, MATH-500, and IFEval, and pass@1 on HumanEval and MBPP. Decoding efficiency is measured by effective tokens per second (TPS), defined as the number of postprocessed output tokens divided by wall-clock decoding time. All experiments are conducted on NVIDIA RTX 4090 GPUs.

4.2 MAIN RESULTS

Overall performance. Table 1 reports task performance and decoding efficiency across five benchmarks on LLaDA and Dream. D2D achieves the highest TPS in every model–benchmark setting while matching or exceeding Standard decoding in task performance, yielding 2.19–$9.92\times$ speedups over Standard and 1.21–$1.64\times$ over Confidence decoding. These gains consistently span both models and all three task categories. Among the competing baselines, DCD achieves the highest average throughput on both models while maintaining comparable task performance. Nevertheless, D2D further improves TPS over DCD by 15.6–42.2% on LLaDA and 10.2–25.8% on Dream. This includes a simultaneous accuracy gain on LLaDA GSM8K. On Dream, D2D also closely matches the average task scores of Confidence and DCD while providing higher throughput across all five benchmarks. Overall, D2D improves decoding efficiency consistently across model architectures and task categories without sacrificing task performance relative to Standard decoding.

Performance across generation lengths. We evaluate D2D with maximum generation lengths of 128, 256, and 512 tokens on GSM8K and HumanEval using LLaDA and Dream. As shown in Table 2, D2D consistently achieves the highest TPS across all 12 configurations, with up to $1.52\times$

Table 1: Main results on five benchmarks with LLaDA and Dream. **Bold** indicates the best task performance and highest TPS.

Benchmark	Method	LLaDA-8B-Instruct			Dream-v0-Instruct-7B		
		Acc/Pass@1	TPS	Speedup	Acc/Pass@1	TPS	Speedup
GSM8K (5-shot)	Standard	77.18	5.28	1.00×	80.52	3.60	1.00×
	Confidence	76.35 (−0.83)	17.02	3.22×	80.59 (+0.07)	13.67	3.80×
	KLASS	76.95 (−0.23)	12.14	2.30×	**81.05** (+0.53)	7.06	1.96×
	WINO	74.53 (−2.65)	15.78	2.99×	77.18 (−3.34)	15.40	4.28×
	DCD	77.33 (+0.15)	17.69	3.35×	80.67 (+0.15)	16.23	4.51×
	D2D	**78.24** (+1.06)	**25.15**	**4.76×**	80.67 (+0.15)	**18.34**	**5.09×**
MATH500 (4-shot)	Standard	38.40	7.27	1.00×	45.20	6.13	1.00×
	Confidence	**38.80** (+0.40)	18.06	2.48×	46.20 (+1.00)	16.47	2.69×
	KLASS	36.80 (−1.60)	13.34	1.83×	**46.80** (+1.60)	10.26	1.67×
	WINO	37.80 (−0.60)	15.30	2.10×	45.40 (+0.20)	18.31	2.99×
	DCD	37.80 (−0.60)	18.67	2.57×	45.80 (+0.60)	17.88	2.92×
	D2D	**38.80** (+0.40)	**24.53**	**3.37×**	46.60 (+1.40)	**21.38**	**3.49×**
HumanEval (0-shot)	Standard	**45.12**	7.43	1.00×	56.71	7.38	1.00×
	Confidence	**45.12** (±0.00)	16.31	2.20×	59.15 (+2.44)	22.65	3.07×
	KLASS	44.51 (−0.61)	10.96	1.48×	**60.98** (+4.27)	13.90	1.88×
	WINO	43.90 (−1.22)	15.96	2.15×	57.93 (+1.22)	23.21	3.14×
	DCD	**45.12** (±0.00)	15.98	2.15×	59.15 (+2.44)	25.15	3.41×
	D2D	**45.12** (±0.00)	**22.36**	**3.01×**	58.54 (+1.83)	**31.65**	**4.29×**
MBPP (3-shot)	Standard	39.20	2.07	1.00×	55.20	1.71	1.00×
	Confidence	39.20 (±0.00)	13.12	6.34×	55.60 (+0.40)	10.32	6.04×
	KLASS	38.60 (−0.60)	8.78	4.24×	56.00 (+0.80)	5.98	3.50×
	WINO	39.60 (+0.40)	7.22	3.49×	55.20 (±0.00)	9.25	5.41×
	DCD	**40.00** (+0.80)	16.19	7.82×	**56.80** (+1.60)	15.17	8.87×
	D2D	39.20 (±0.00)	**18.92**	**9.14×**	56.20 (+1.00)	**16.97**	**9.92×**
IFEval (0-shot)	Standard	56.75	14.70	1.00×	55.27	14.03	1.00×
	Confidence	56.75 (±0.00)	26.97	1.83×	**56.19** (+0.92)	25.50	1.82×
	KLASS	**56.93** (+0.18)	22.39	1.52×	53.79 (−1.48)	21.22	1.51×
	WINO	56.19 (−0.56)	27.62	1.88×	51.20 (−4.07)	26.94	1.92×
	DCD	56.75 (±0.00)	28.42	1.93×	55.27 (±0.00)	27.91	1.99×
	D2D	**56.93** (+0.18)	**32.86**	**2.24×**	55.64 (+0.37)	**30.75**	**2.19×**
Avg	Standard	51.33	7.35	1.00×	58.58	6.57	1.00×
	Confidence	51.24 (−0.09)	18.30	2.49×	59.55 (+0.97)	17.72	2.70×
	KLASS	50.76 (−0.57)	13.52	1.84×	**59.72** (+1.14)	11.68	1.78×
	WINO	50.40 (−0.93)	16.38	2.23×	57.38 (−1.20)	18.62	2.83×
	DCD	51.40 (+0.07)	19.39	2.64×	59.54 (+0.96)	20.47	3.12×
	D2D	**51.66** (+0.33)	**24.76**	**3.37×**	59.53 (+0.95)	**23.82**	**3.63×**

speedup over Confidence decoding and $1.45\times$ over DCD. Task performance remains comparable to Confidence decoding across lengths, with increases up to 1.89% and acceptable loss. These results demonstrate the robustness of D2D's quality–efficiency trade-off across generation lengths. We use a maximum generation length of 256 tokens in other experiments unless otherwise specified.

4.3 Ablation Study and Further Analysis

Hyperparameter sensitivity. We examine the joint commitment budget δ and the Shell admission parameter ϵ_A on LLaDA using GSM8K and HumanEval, varying one parameter while fixing the other. As shown in Figure 3, increasing δ from 0.2 to 0.6 steadily improves TPS, while task performance peaks at $\delta = 0.4$ on both benchmarks. Larger budgets enable more aggressive parallel commitment, trading generation quality for higher throughput. The effect of ϵ_A is more task-dependent: GSM8K accuracy remains near or above Standard decoding across $\epsilon_A \in [0.5, 0.9]$ and peaks at 0.9, whereas HumanEval pass@1 remains stable over $\epsilon_A \in [0.5, 0.7]$ before declining at larger values. Compared with δ, varying ϵ_A produces smaller changes in TPS, while $\epsilon_A = 1.0$ substantially reduces quality on both tasks. Accordingly, we use $\delta = 0.4$ for both benchmarks and set $\epsilon_A = 0.9$ for GSM8K and $\epsilon_A = 0.7$ for HumanEval, as marked by stars in Figure 3.

Table 2: Generation-length robustness on LLaDA and Dream.**Bold** indicates the best task performance and highest TPS.

Benchmark	Generation Length	Method	LLaDA-8B-Instruct			Dream-v0-Instruct-7B		
			Acc/Pass@1	TPS	Speedup	Acc/Pass@1	TPS	Speedup
GSM8K (5-shot)	128	Standard	75.89	6.06	1.00×	70.36	6.61	1.00×
		Confidence	75.97	15.87	2.62×	**70.74**	15.64	2.37×
		DCD	76.04	15.86	2.62×	69.45	16.00	2.42×
		D2D	**76.27**	**22.92**	**3.78×**	70.28	**19.05**	**2.88×**
	256	Standard	77.18	5.28	1.00×	80.52	3.60	1.00×
		Confidence	76.35	17.02	3.22×	80.59	13.67	3.80×
		DCD	77.33	17.69	3.35×	**80.67**	16.23	4.51×
		D2D	**78.24**	**25.15**	**4.76×**	**80.67**	**18.34**	**5.09×**
	512	Standard	76.50	2.47	1.00×	**79.76**	1.40	1.00×
		Confidence	**77.71**	13.40	5.43×	79.53	8.85	6.32×
		DCD	**77.71**	15.15	6.13×	79.61	12.71	9.08×
		D2D	76.57	**19.04**	**7.71×**	79.53	**13.49**	**9.64×**
HumanEval (0-shot)	128	Standard	40.85	13.69	1.00×	55.49	10.99	1.00×
		Confidence	39.63	24.90	1.82×	**58.54**	28.63	2.61×
		DCD	40.85	24.59	1.80×	**58.54**	29.89	2.72×
		D2D	**41.46**	**32.66**	**2.39×**	57.93	**40.74**	**3.71×**
	256	Standard	**45.12**	7.43	1.00×	56.71	7.38	1.00×
		Confidence	**45.12**	16.31	2.20×	**59.15**	22.65	3.07×
		DCD	**45.12**	15.98	2.15×	**59.15**	25.15	3.41×
		D2D	**45.12**	**22.36**	**3.01×**	58.54	**31.65**	**4.29×**
	512	Standard	43.29	2.79	1.00×	**54.27**	2.78	1.00×
		Confidence	41.46	7.58	2.72×	53.05	14.96	5.38×
		DCD	**43.90**	7.26	2.60×	53.66	20.87	7.51×
		D2D	43.29	**9.79**	**3.51×**	**54.27**	**22.29**	**8.02×**

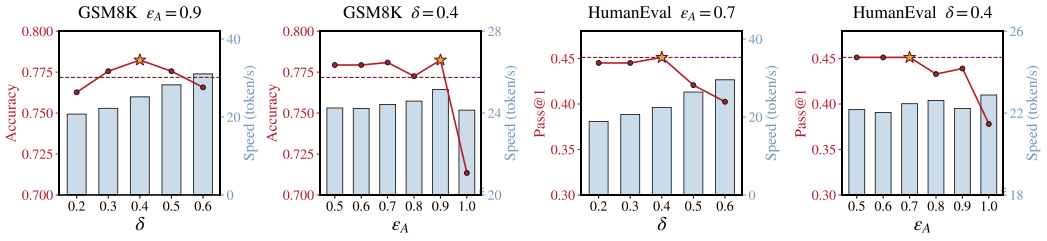

Figure 3: Accuracy–speed trade-off of D2D under different δ and ϵ_A. Red curves show GSM8K accuracy or HumanEval Pass@1, and blue bars show wall-clock TPS. Dark-red dashed lines denote standard decoding quality. Stars mark the main experiment settings.

Comparison of Core selection strategies. To examine how Core selection affects the quality–efficiency trade-off, we compare Confidence selection, Factor selection (Wu et al., 2026), Fréchet profile selection (Kasa et al., 2026), and our budgeted Core on LLaDA using GSM8K. All variants share the same Shell admission and deferred verification procedure and differ only in the Core selector; for each selector, we sweep its relevant hyperparameters and report the configuration with the best quality–efficiency trade-off, thereby isolating the effect of Core selection. As shown in Figure 4, our budgeted Core lies in the upper-right region, combining high throughput with the highest accuracy among the compared variants. Confidence selection preserves accuracy above the Standard decoding baseline but yields lower throughput, whereas Factor and Fréchet profile selection achieve slightly higher throughput at the cost of accuracy below the Standard baseline. These results show that Core selection remains important even under the same deferred verification mechanism: our budgeted selector enables greater parallelism than Confidence selection while preserving higher quality than the more aggressive alternatives.

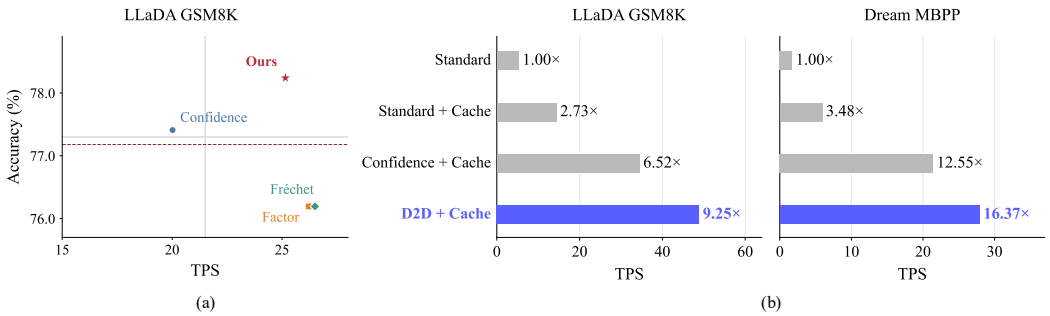

Figure 4: (a) Core selection comparison with the same Shell mechanism on LLaDA GSM8K; the dashed line marks Standard decoding accuracy baseline. (b) KV caching results on LLaDA GSM8K and Dream MBPP, with speedup relative to uncached Standard decoding.

Table 3: Ablation study of different components.

Benchmark	Variant	Acc/Pass@1	TPS
GSM8K (5-shot)	Confidence	76.35	17.02
	+ Identity-only	74.45	22.29
	+ Joint	77.41	20.53
	+ Joint + Core (**D2D**)	**78.24**	**25.15**
HumanEval (0-shot)	Confidence	**45.12**	16.31
	+ Identity-only	35.98	**24.50**
	+ Joint	42.68	21.11
	+ Joint + Core (**D2D**)	**45.12**	22.36

Component ablation. Table 3 disentangles the effects of Shell-based deferred verification and budgeted Core selection on LLaDA using GSM8K and HumanEval. Starting from Confidence decoding with $\tau = 0.9$, we first add a Shell and verify deferred proposals using token-identity consistency alone. This *identity-only* variant improves throughput but degrades task performance, especially on HumanEval, where pass@1 drops by 9.14 percentage points, showing that token persistence alone is insufficient for reliable verification. Our *joint* verification additionally requires non-decreasing confidence under updated context, largely recovering quality while retaining higher throughput than Confidence. This shows that adding the Shell with deferred verification can effectively exploit proposals beyond the Confidence commitment rule. Finally, replacing Confidence-based Core selection with our budgeted Core further improves quality and throughput under the Shell verification mechanism, yielding a better quality–efficiency trade-off for immediate commitment. Full D2D achieves 47.8% and 37.1% higher TPS than Confidence decoding on GSM8K and HumanEval, respectively, while improving GSM8K accuracy and preserving HumanEval pass@1. Together, these results validate the complementary benefits of Shell-based deferred verification and budgeted Core selection.

Compatibility with KV caching. D2D increases decoding progress per forward pass, while KV caching reduces the computational cost of each pass. To exploit these complementary mechanisms, we integrate D2D with the prefix KV cache from Fast-dLLM (Wu et al., 2026) and evaluate the combination on LLaDA using GSM8K and Dream using MBPP. As shown in Figure 4, cached D2D achieves up to $16.37\times$ speedup over uncached standard decoding and retains a throughput advantage over cached confidence decoding with acceptable accuracy loss. This advantage shows that computational reuse leaves further room for improving token commitment: D2D continues to advance decoding more efficiently when caching is enabled, these results demonstrate that D2D combines effectively with KV caching.

5 Conclusion

We presented Decouple to Decode (D2D), a training-free framework that separates immediate commitment from deferred verification in diffusion language models. D2D commits reliable predictions through a budgeted Core while retaining candidates in a masked Shell for verification. Shell proposals are re-evaluated in the next forward pass using token-identity consistency and non-decreasing confidence, with acceptance constrained by the commitment budget. This design exploits cross-step evidence without additional model evaluations or exposing unverified proposals to the decoding context. Experiments on LLaDA and Dream across five benchmarks demonstrate up to $9.92\times$ wall-clock speedup over standard decoding while maintaining comparable task performance, reaching $16.37\times$ with KV caching. Overall, our results further demonstrate the value of decoupling immediate commitment from selective verification for efficient dLLM decoding.

AI USE STATEMENT

Our use of generative AI was limited to writing assistance, including language polishing, grammar correction, and refinement of wording and sentence structure to improve the clarity, coherence, and readability of the manuscript. The authors take full responsibility for the final content of this paper, including its scientific claims and conclusions.

ETHICS STATEMENT

D2D studies training-free decoding acceleration for diffusion language models using existing models and standard benchmarks. Improved inference efficiency can make these models more accessible, but may also lower the cost of generating misleading or harmful content. D2D focuses on decoding efficiency and does not address the underlying models' potential biases, factual errors, or safety limitations. Its deployment should therefore be accompanied by application-specific safety evaluation and appropriate safeguards, with attention to the intended use and potential misuse of generated content.

REPRODUCIBILITY STATEMENT

We describe the Core selection, Shell admission, and deferred verification procedures in Section 3.2, with theoretical assumptions and proofs provided in Appendix B. Section 4.1 specifies the evaluated models and benchmarks, decoding settings, evaluation framework, performance metrics, and hardware configuration. Additional hyperparameter settings for D2D and the baselines are provided in Appendix C. Together, these descriptions document the algorithm and experimental configurations needed to reproduce our evaluation.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021a.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021b.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas S Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *Advances in Neural Information Processing Systems*, 38:55981–56007, 2026.

Kecheng Chen, Ziru Liu, Xijia Tao, Hui Liu, Xinyu Fu, Suiyun Zhang, Dandan Tu, Lingpeng Kong, Rui Liu, and Haoliang Li. Beyond confidence: Adaptive and coherent decoding for diffusion language models. *arXiv preprint arXiv:2512.02044*, 2025.

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

Yuzhe Chen, Jiale Cao, Xuyang Liu, Jin Xie, Aiping Yang, and Yanwei Pang. Stdec: Spatio-temporal stability guided decoding for dllms. *arXiv preprint arXiv:2604.06330*, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems, 2021. *URL https://arxiv. org/abs/2110.14168*, 9, 2021.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 6112–6121, 2019.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Feng Hong, Geng Yu, Yushi Ye, Haicheng Huang, Huangjie Zheng, Ya Zhang, Yanfeng Wang, and Jiangchao Yao. Wide-in, narrow-out: Revokable decoding for efficient and effective dllms. In *International Conference on Learning Representations*, volume 2026, pp. 40132–40150, 2026.

Siva Rajesh Kasa, Yasong Dai, Sumit Negi, and Hongdong Li. Fast-dllm++: Fr\'{e} chet profile decoding for faster diffusion llm inference. *arXiv preprint arXiv:2606.02955*, 2026.

Seo Hyun Kim, Sunwoo Hong, Hojung Jung, Youngrok Park, and Se-Young Yun. Klass: Kl-guided fast inference in masked diffusion models. *Advances in Neural Information Processing Systems*, 38:92267–92301, 2026.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1173–1182, 2018.

Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Soroush Vosoughi, and Shiwei Liu. Diffusion language models know the answer before decoding. *arXiv preprint arXiv:2508.19982*, 2025.

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

Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dkv-cache: The cache for diffusion language models. *Advances in Neural Information Processing Systems*, 38:149009–149033, 2026.

Amr Mohamed, Guokan Shang, and Michalis Vazirgiannis. Less is more: Mutual-stability sampling for diffusion language models. *arXiv preprint arXiv:2606.16908*, 2026.

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. *Advances in Neural Information Processing Systems*, 38:50608–50646, 2026.

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.

Yingte Shu, Yuchuan Tian, Chao Xu, Yunhe Wang, and Hanting Chen. Deferred commitment decoding for diffusion language models. *arXiv preprint arXiv:2601.02076*, 2026.

Kangyu Wang, Zhiyun Jiang, Haibo Feng, Weijia Zhao, Lin Liu, Jianguo Li, Zhenzhong Lan, and Weiyao Lin. Creditdecoding: Accelerating parallel decoding in diffusion large language models with trace credit. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11105–11123, 2026.

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.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Shaorong Zhang, Longxuan Yu, Rob Brekelmans, Luhan Tang, Salman Asif, and Greg Ver Steeg. Generation order and parallel decoding in masked diffusion models: An information-theoretic perspective. *arXiv preprint arXiv:2602.00286*, 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.

A D2D ALGORITHM

Algorithm 1 One Decoding Step of Decouple to Decode (D2D)

Require: Model p_θ, prompt P, sequence y_t, masked positions U_t, Shell buffer Q_{t-1}, budgets δ, ϵ_A
Ensure: Updated sequence y_{t+1} and Shell buffer Q_t

Let $A^{(k)}$ denote the first k positions of ordered A, with $A^{(0)} = \varnothing$.

$p_t \leftarrow p_\theta(\cdot \mid y_t, P)$
$\hat{x}_{t,i} \leftarrow \arg\max_v p_{t,i}(v)$, $c_{t,i} \leftarrow p_{t,i}(\hat{x}_{t,i})$ for all $i \in U_t$
$F_t(S) \leftarrow 1 - \prod_{i \in S} c_{t,i}$, with $F_t(\varnothing) = 0$
$\widetilde{V}_t \leftarrow \{i : (i, x, c) \in Q_{t-1},\ i \in U_t,\ \hat{x}_{t,i} = x,\ c_{t,i} \geq c\}$ ▷ Shell verification
Sort $\widetilde{V}_t$ by decreasing $c_{t,i}$
$k_V \leftarrow \max\{k \in \{0, \ldots, |\widetilde{V}_t|\} : F_t(\widetilde{V}_t^{(k)}) \leq \delta\}$; $V_t \leftarrow \widetilde{V}_t^{(k_V)}$ ▷ Budget truncation
$E_t \leftarrow U_t \setminus V_t$, sorted by decreasing $c_{t,i}$ ▷ Core selection
$k_C \leftarrow \max\{k \in \{0, \ldots, |E_t|\} : F_t(V_t \cup E_t^{(k)}) \leq \delta\}$; $C_t \leftarrow E_t^{(k_C)}$
$H_t \leftarrow \varnothing$; $a_t \leftarrow \epsilon_A[1 - F_t(V_t) - F_t(C_t)]$ ▷ Shell selection
if $a_t > 0$ **then**
    $k_A \leftarrow \max\{k \in \{k_C, \ldots, |E_t|\} : F_t(E_t^{(k)} \setminus C_t) \leq a_t\}$
    $H_t \leftarrow E_t^{(k_A)} \setminus C_t$
end if
$S_t \leftarrow V_t \cup C_t$
if $S_t = \varnothing$ and $U_t \neq \varnothing$ **then**
    $i^\star \leftarrow \arg\max_{i \in U_t} c_{t,i}$; $S_t \leftarrow \{i^\star\}$; $H_t \leftarrow H_t \setminus \{i^\star\}$
end if
$y_{t+1} \leftarrow y_t$; $y_{t+1,i} \leftarrow \hat{x}_{t,i}$ for all $i \in S_t$
$Q_t \leftarrow \{(i, \hat{x}_{t,i}, c_{t,i}) : i \in H_t\}$
return y_{t+1}, Q_t

B PROOFS AND ADDITIONAL ANALYSIS

B.1 PROBABILISTIC FORMULATION AND PREFIX OPTIMALITY

Conditional selection model. Fix iteration t and condition on the selection evidence $\mathcal{D}_t$, which determines U_t, V_t, E_t, the current proposals, and their confidence scores. Let $Y_{t,i} = \mathbf{1}\{X_i^\star = \hat{x}_{t,i}\}$ and $q_{t,i} = \Pr(Y_{t,i} = 1 \mid \mathcal{D}_t)$. We assume $c_{t,i} \in (0, 1]$ for $i \in U_t$ and deterministic tie-breaking. Under $q_{t,i} = c_{t,i}$ and conditional joint independence of $(Y_{t,i})_{i \in U_t}$ given $\mathcal{D}_t$, any $\mathcal{D}_t$-measurable selection $S \subseteq E_t$ satisfies

$$\mathbb{E}\left[\sum_{i \in S} Y_{t,i} \,\middle|\, \mathcal{D}_t\right] = \sum_{i \in S} c_{t,i}, \qquad \Pr(\exists i \in V_t \cup S : Y_{t,i} = 0 \,|\, \mathcal{D}_t) = F_t(V_t \cup S). \tag{10}$$

For $\rho < 1$, positivity of the confidences further gives

$$F_t(V_t \cup S) \leq \rho \quad \Longleftrightarrow \quad \sum_{i \in V_t \cup S} -\log c_{t,i} \leq -\log(1 - \rho). \tag{11}$$

Thus each candidate contributes reward $c_{t,i}$ and additive risk cost $-\log c_{t,i}$. The score-based optimization itself remains well defined without the posterior assumptions; these assumptions identify it with the conditional objective and constraint in Eq. (6).

Proof of Theorem 3.1. Let k_ρ be the largest index satisfying $F_t(V_t \cup E_t^{(k_\rho)}) \leq \rho$. For any feasible $S \subseteq E_t$ with $|S| = m$, replacing lower-confidence selected positions with higher-confidence unselected ones gives

$$\sum_{i \in E_t^{(m)}} c_{t,i} \geq \sum_{i \in S} c_{t,i}, \qquad \prod_{i \in E_t^{(m)}} c_{t,i} \geq \prod_{i \in S} c_{t,i}. \tag{12}$$

Since V_t is fixed and disjoint from E_t, the product inequality implies that $E_t^{(m)}$ is feasible whenever S is feasible. Hence $m \le k_\rho$, and Eq. (12) further shows that $E_t^{(k_\rho)}$ attains no smaller objective than any feasible S. Therefore the largest feasible prefix solves Eq. (6) and has maximum feasible cardinality. Finally, if $\rho_1 \le \rho_2$, every prefix feasible at ρ_1 remains feasible at ρ_2, so the fixed-order prefix solutions are nested.

Progress fallback. When $V_t \cup C_t = \varnothing$, let $i^\star = \arg\max_{i \in U_t} c_{t,i}$. Since even the highest-confidence singleton is infeasible, $F_t(\{i^\star\}) > \delta$. For any nonempty $S \subseteq U_t$,

$$F_t(S) = 1 - \prod_{i \in S} c_{t,i} \ge 1 - c_{t,i^\star} = F_t(\{i^\star\}).$$

Hence no nonempty commitment satisfies the original budget, and $\{i^\star\}$ is the minimum-risk nonempty choice.

B.2 Equivalence to a Relaxed Joint-Confidence Constraint

Let C_t be the Core from Eq. (3), $S_t = V_t \cup C_t$, and $A_t = C_t \cup H_t$. Write the Shell allowance as

$$b_t = \epsilon_A \max\{1 - F_t(V_t) - F_t(C_t), 0\}. \tag{13}$$

Let $u = \prod_{i \in V_t} c_{t,i}$ and $v = \prod_{i \in C_t} c_{t,i}$, so $b_t = \epsilon_A[u + v - 1]_+$, where $[x]_+ = \max\{x, 0\}$. Since Core selection ensures $F_t(S_t) \le \delta$, we have $uv = 1 - F_t(S_t) \ge 1 - \delta$. Using $2\sqrt{uv} \le u + v \le 1 + uv$ for $u, v \in (0, 1]$ gives

$$\epsilon_A[2\sqrt{1-\delta} - 1]_+ \le b_t \le \epsilon_A[1 - F_t(S_t)] \le \epsilon_A. \tag{14}$$

For any H disjoint from S_t,

$$F_t(S_t \cup H) = F_t(S_t) + [1 - F_t(S_t)]F_t(H). \tag{15}$$

Define $\beta_t = F_t(S_t) + [1 - F_t(S_t)]b_t$. Then for every $k \ge k_C$,

$$F_t(E_t^{(k)} \setminus C_t) \le b_t \quad \Longleftrightarrow \quad F_t(V_t \cup E_t^{(k)}) \le \beta_t. \tag{16}$$

Thus Shell admission is exactly a joint-risk constraint on the extended confidence-ranked prefix.

If $\beta_t \ge \delta$, Eq. (16) directly makes A_t the largest feasible prefix at budget β_t. If $\beta_t < \delta$, Core maximality implies that no prefix beyond C_t is feasible at β_t, so $H_t = \varnothing$ and replacing β_t by δ leaves the selected prefix unchanged. Hence, in all cases,

$$(1 + \epsilon_t)\delta = \max\{\delta, \beta_t\}, \qquad A_t = E_t^{(k_A)} \tag{17}$$

is the largest feasible prefix at the relaxed budget $(1+\epsilon_t)\delta$. By Theorem 3.1, C_t and A_t are therefore optimal for Eq. (6) at $\rho = \delta$ and $\rho = (1 + \epsilon_t)\delta$, respectively. Equivalently,

$$F_t(V_t \cup A_t) \le (1 + \epsilon_t)\delta \quad \Longleftrightarrow \quad \prod_{i \in V_t \cup A_t} c_{t,i} \ge 1 - (1 + \epsilon_t)\delta. \tag{18}$$

Here, ϵ_t is induced by the realized Shell admission rule rather than optimized independently. The Shell $H_t = A_t \setminus C_t$ is therefore the augmentation enabled by relaxing the same risk constraint.

B.3 Verification of Fixed Historical Proposals

Proof of Proposition 3.2. Let $\widetilde{V}_{t+1} \subseteq H_t$ denote the proposals passing identity and support verification before budget truncation. For every $i \in \widetilde{V}_{t+1}$,

$$\hat{x}_{t+1,i} = \hat{x}_{t,i}, \qquad c_{t+1,i} = p_\theta^i(\hat{x}_{t,i} \mid y_{t+1}, P) \ge c_{t,i}. \tag{19}$$

After sorting $\widetilde{V}_{t+1}$ by decreasing updated confidence, budget truncation retains its largest prefix V_{t+1} satisfying $F_{t+1}(V_{t+1}) \le \delta$. Hence, for any $R \subseteq V_{t+1}$, the two confidence scores evaluate the same token assignment, and monotonicity of c and $-\log c$ gives

$$\sum_{i \in R} c_{t+1,i} \ge \sum_{i \in R} c_{t,i}, \qquad \sum_{i \in R} -\log c_{t+1,i} \le \sum_{i \in R} -\log c_{t,i}. \tag{20}$$

Moreover, since $V_{t+1} \subseteq \widetilde{V}_{t+1} \subseteq H_t$ and all confidence factors lie in $(0, 1]$,

$$\prod_{i \in V_{t+1}} c_{t+1,i} \geq \prod_{i \in V_{t+1}} c_{t,i} \geq \prod_{i \in H_t} c_{t,i}, \tag{21}$$

which yields

$$F_{t+1}(V_{t+1}) \leq \delta, \qquad F_{t+1}(V_{t+1}) \leq F_t(H_t) \leq b_t. \tag{22}$$

These are deterministic score inequalities and require neither calibration nor independence.

Posterior interpretation. For a stored proposal, define

$$q_{t,i} = \Pr(Y_{t,i} = 1 \mid \mathcal{D}_t), \qquad q^+_{t,i} = \Pr(Y_{t,i} = 1 \mid \mathcal{D}_t, Z_{t+1}).$$

Under the identifications $q_{t,i} = c_{t,i}$ and $q^+_{t,i} = p^i_\theta(\hat{x}_{t,i} \mid y_{t+1}, P)$, Eq. (19) implies $q^+_{t,i} \geq q_{t,i}$ for every retained proposal. Thus, verification preserves or improves the posterior reward of the same historical assignment while not increasing its log-risk cost.

Inheritance of the relaxed admission bound. Consider a budget-controlled admission step t, for which $S_t = V_t \cup C_t$. If the progress fallback is invoked instead, $H_t = \varnothing$ by construction, so no deferred proposal is carried to the next step. For a budget-controlled step, the admitted historical batch is

$$V_t \cup A_t = S_t \cup H_t.$$

Since $V_{t+1} \subseteq H_t$ and verification preserves the same historical token assignments,

$$\{\exists i \in S_t \cup V_{t+1} : Y_{t,i} = 0\} \subseteq \{\exists i \in V_t \cup A_t : Y_{t,i} = 0\}. \tag{23}$$

Conditioning on $\mathcal{D}_t$ and applying the admission-time probability model gives

$$\Pr(\exists i \in S_t \cup V_{t+1} : Y_{t,i} = 0 \mid \mathcal{D}_t) \leq F_t(V_t \cup A_t) \leq (1 + \epsilon_t)\delta. \tag{24}$$

Thus, retained historical assignments inherit the relaxed admission-time bound, while budget truncation independently restores the original budget for the verified set at step $t + 1$.

Budget for subsequent commitment. By construction, $F_{t+1}(V_{t+1}) \leq \delta$, so the next Core selection starts from a feasible verified set. Moreover, retained proposals consume no more log-risk budget under their updated scores than under their admission-time scores, since

$$\sum_{i \in V_{t+1}} \log \frac{c_{t+1,i}}{c_{t,i}} \geq 0. \tag{25}$$

With the verified set and fresh-candidate scores fixed, the remaining budget for fresh Core selection therefore cannot decrease. If the subsequent step admits no nonempty commitment, the separate progress fallback is invoked as described in Section 3.2.

C HYPERPARAMETER SETTINGS

We follow the settings recommended in the original papers or official implementations. Standard unmasks the highest confidence position per step. Confidence uses a threshold of 0.9 following the settings in Fast-DLLM. DCD uses an initial window size of 16, a maximum window size of 128, and a decoding threshold of 0.9. For KLASS, we fix history_length to 2, with confidence thresholds selected from $\{0.6, 0.7, 0.8, 0.9\}$ and KL thresholds from $\{0.001, 0.005, 0.01, 0.015\}$. For WINO, we use drafting thresholds from $\{0.6, 0.7, 0.8\}$ and verification threshold 0.9.

We specify D2D configurations as parameter pairs (δ, ϵ_A) for each model. On LLaDA, we use $(0.4, 0.7)$ for MATH500, HumanEval, $(0.4, 0.9)$ for GSM8K, IFEval, and $(0.45, 0.5)$ for MBPP. On Dream, we use $(0.3, 0.7)$ for GSM8K, MATH500, $(0.4, 0.9)$ for Humaneval, $(0.4, 0.8)$ for IFEval and $(0.4, 0.4)$ for MBPP.

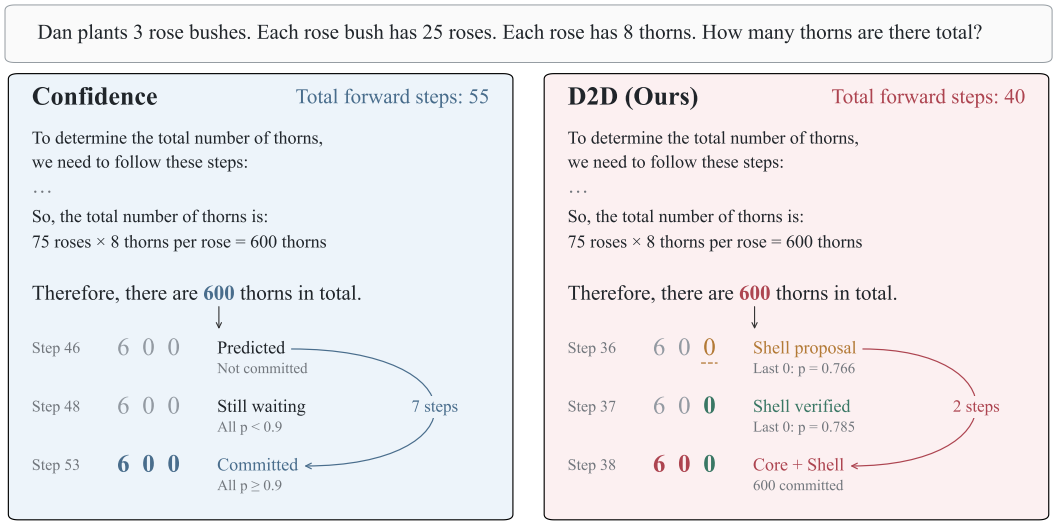

Figure 5: Case study on LLaDA using GSM8K.

D CASE STUDY

Figure 5 illustrates how D2D commits the same correct answer earlier than Confidence decoding while producing an identical response on LLaDA using GSM8K. Confidence first predicts 600 at step 46 but leaves all three tokens uncommitted because their confidence scores remain below 0.9. It commits the complete answer at step 53, 7 steps later. D2D proposes the final 0 through Shell at step 36 with confidence 0.766. At step 37, the token remains unchanged and its confidence increases to 0.785, allowing it to pass verification despite remaining below 0.9. At step 38, Core commits 6 and the first 0, completing the answer 2 steps after the proposal. Verification uses the next regular forward pass without an additional model call. Overall, D2D requires 40 forward passes versus 55 for Confidence with a 27.3% reduction. This demonstrates earlier token commitment and fewer forward passes, rather than a measured wall-clock speedup.