CONGO: CONSISTENCY-GUIDED OPTIMIZATION FOR PARALLEL DECODING IN DISCRETE DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete diffusion is an emerging paradigm for generative AI, in which a sequence of tokens is generated through iterative updates using a learned denoising model. While the standard formulation randomly selects token positions to update in each denoising iteration, common heuristics select the positions with highest marginal confidence or minimum inter-token dependency to increase generation quality. These approaches, however, strictly isolate the position selection from the content selection. Such *value-blind* position selection hinders the commitment of jointly consistent tokens, which can be indicated by explicit constraints or learned priors. In this work, we propose consistency-guided optimization (ConGO) for parallel decoding in discrete diffusion models, which formulates the update selection as a quadratic unconstrained binary optimization problem. Guided by a lightweight, *value-aware consistency model*, ConGO offers a flexible spectrum of integration depending on the reliability of the available knowledge. When strong, explicit domain knowledge is available, such as in Sudoku, ConGO *jointly* selects both the positions and their content, achieving significantly higher accuracy on hard problems (up to a 42.4 pp improvement). When explicit constraints are absent, ConGO restricts its intervention purely to position selection based on learned pairwise priors (e.g., bigrams). Integrated into both marginal and dependency-aware decoding of LLaDA-8B-Instruct in coding, ConGO improves each method's accuracy-NFE trade-off towards fewer function evaluations, with the dependency-aware integration establishing a novel Pareto front.

1 INTRODUCTION

While autoregressive (AR) models remain the commercial standard for general-purpose LLMs, non-AR iterative discrete diffusion models (DDMs) (Austin et al., 2021a; Campbell et al., 2022; Lou et al., 2024; Sahoo et al., 2024; Ou et al., 2025; Shi et al., 2024; Rütte et al., 2025) show advantages in optimization (Sun et al., 2026; Feng et al., 2026; Bai et al., 2026), self-correction (Bie et al., 2026; Wang et al., 2026), and data-efficient training (Prabhudesai et al., 2025; Rütte et al., 2026; Sahoo et al., 2026). Besides enabling new generation capabilities, DDM's main advantage is the potential for sub-linear inference latency, thanks to committing multiple tokens per model pass. For example, the current state-of-the-art (SoTA) open-source DDM, DiffusionGemma (DiffusionGemma Team et al., 2026), commits roughly 20 tokens per model pass, thereby generating 1500 tokens per second on an NVIDIA H100 GPU in single-user serving.

During generation, DDMs iteratively refine a sequence by replacing noisy tokens, which are either represented with a mask token (masked diffusion model; MDM) or with a randomly selected token (uniform diffusion model; UDM). While the standard DDM formulation would randomly select the update position and sample the content from the denoising model's distribution, practical solutions make deterministic decisions to achieve higher generation quality and parallelism. This is mainly driven by the modeling of the output distribution, which approximates the conditional joint probability as the product of the per-token marginals (Liu et al., 2025; Bansal and Sanghavi, 2025; Sun et al., 2025; Jin et al., 2025; Kang et al., 2026; Feng et al., 2025; Zhong et al., 2026; Hersche et al., 2026). Independently sampling a set of tokens from this approximate distribution is benign if the selected tokens are near-independent. However, when target tokens exhibit strong cross-correlations, independent sampling frequently selects candidates that are individually likely but globally incompatible. This joint inconsistency appears across multiple domains such as natural

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.

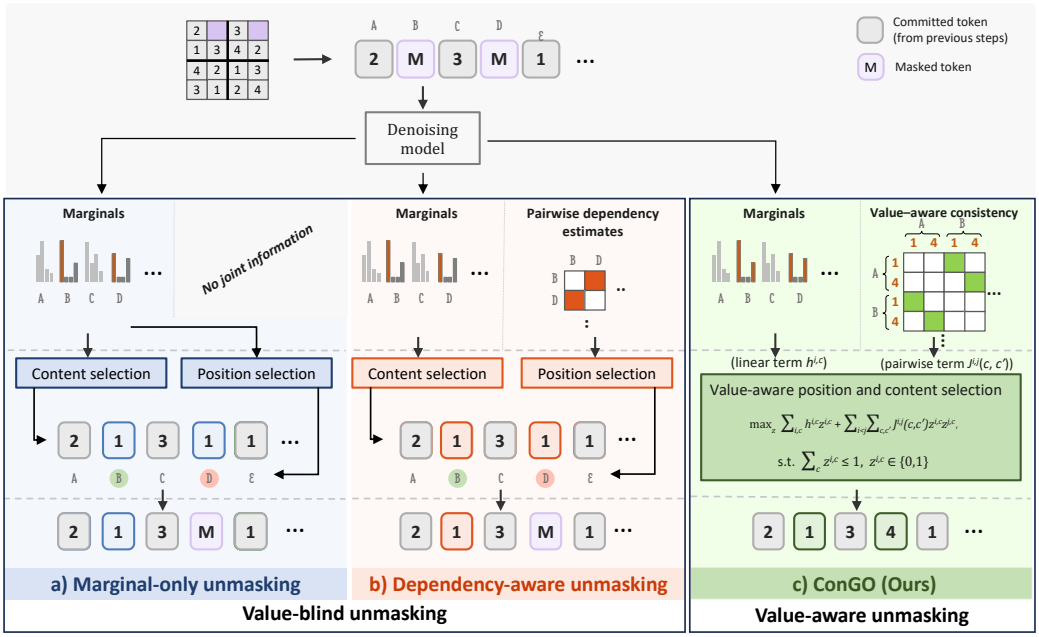

Figure 1: Comparison of ConGO against existing update strategies, applied to a masked diffusion model for a 4×4 Sudoku. The two masked cells B and D share the row; hence, they are strongly coupled and can be jointly determined. The denoiser narrows the choice down to $\{1, 4\}$ for both. (a) Marginal unmasking relies solely on per-position scores. This can result in committing both positions in the same step with potentially inconsistent values, or deferring one of the updates to the next step. (b) Dependency unmasking incorporates pairwise position dependencies. Since the two positions are identified as dependent, it defers committing the second position for the next step. (c) ConGO utilizes marginals and pairwise, value-aware consistency terms to *jointly* select both the position and the value. By solving this binary optimization problem, ConGO can safely commit coupled tokens at compatible values in one step, therefore enabling more parallel updates per step.

language (e.g., two parts of a multi-token entity), code generation (e.g., variable definition and its later use), or biological sequences (e.g., complementary base pairs in RNA structures).

To mitigate errors from such a conditional independence bias, practical approaches take conservative strategies (see Figure 1): they either rely on per-position marginals to commit highly confident tokens (Chang et al., 2022; Wu et al., 2026; Ben-Hamu et al., 2025; Kim et al., 2025a;b), or directly estimate pairwise dependencies to strictly commit uncorrelated positions (Kim et al., 2026; Ringel et al., 2026). Crucially, all these methods strictly isolate position selection from content selection. Such *value-blind* position selection focuses exclusively on deciding *where* to update, fully delegating *what* to the model's isolated marginals. Because they rely on a single scalar per position or pair, it becomes impossible to jointly commit correlated but mutually consistent tokens.

In this work, we introduce consistency-guided optimization (ConGO), which frames update set selection in DDMs as a *value-aware* optimization problem that jointly selects which positions to commit and at which values (see Figure 1). It maximizes a quadratic objective, where the linear terms correspond to the model's marginal confidence and the quadratic terms to value-level interactions between pairs of positions and values, enabling the objective to encourage committing jointly consistent values at multiple positions. The proposed value-aware formulation offers a flexible spectrum of integration. Depending on the reliability of the consistency model, we can control the extent of the intervention, ranging from full *joint selection of positions and content* when backed by exact knowledge, to *value-aware position selection* when relying on softer, learned priors. Moreover, in the absence of a consistency model, ConGO can encompass value-blind marginal and dependency-aware update selection methods. While this paper formulates ConGO for optimizing unmasking schedules in MDMs, it can be applied to the update selection in UDMs as well (see App. D.1.2 for experiments on DiffusionGemma).

Our contributions are as follows:

- We formulate update set selection in parallel decoding as a joint optimization over *which* positions to commit and at *which* values, combining linear terms (reflecting the model's marginal confidence) with quadratic terms (value-level consistency between pairs of positions). This reduces to a quadratic unconstrained binary optimization (QUBO). The reliability of the available consistency model determines a spectrum of integration levels.
- When a strong, explicit consistency model is available (e.g., symbolic constraints), ConGO jointly optimizes over both positions and values. Using Sudoku as a testbed, we show that selecting from the top-3 candidate values improves the accuracy of a 5.4M-parameter MDM by up to 42.4 percentage points (pp) on out-of-distribution problems, compared to both SoTA marginal unmasking schedules and greedy symbolic baselines. Furthermore, ConGO seamlessly transfers to UDMs, improving DiffusionGemma on Sudoku as well.
- When only a weaker value-aware consistency model is available, ConGO can be reduced to only optimizing which positions to commit. Under a first-order Markovian dependence assumption, we show that our formulation is equivalent to committing as many positions as possible while keeping them jointly probable. Additionally, we propose an extension to dependency-aware decoding, ensuring that dependent but consistent tokens can be co-committed. Experimentally, we show that enhancing both marginal and dependency-aware unmasking schedules with a lightweight bigram model over adjacent tokens improves the NFE-accuracy Pareto front of LLaDA-8B-Instruct in coding (HumanEval and MBPP).

2 PRELIMINARIES

This section provides a brief introduction to MDMs; we refer to the seminal works (Austin et al., 2021a; Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2025) for a complete exposition.

2.1 NOTATION

Superscripts index positions in a sequence and subscripts index time. We write $x = (x^1, \dots, x^L)$ for a sequence of length L over a vocabulary $\mathcal{V}$, and $\mathbf{m} \notin \mathcal{V}$ for a dedicated mask token, so that intermediate states in the diffusion process live in $(\mathcal{V} \cup \{\mathbf{m}\})^L$. The state at time $t \in [0, 1]$ is $x_t = (x_t^1, \dots, x_t^L)$, with x_0 a clean sequence and x_1 the fully masked (noisy) one. For a state x_t, we write $\mathcal{M}_t = \{i : x_t^i = \mathbf{m}\}$ for the set of masked positions, and $|\mathcal{M}_t|$ for its size. For $S \subseteq \mathcal{M}_t$ we write x_t^S for the restriction of x_t to S. Finally, $\Delta := \{v \in [0, 1]^{|\mathcal{V}|} : \sum_i v_i = 1\}$ is the simplex over $\mathcal{V}$.

2.2 MASKED DIFFUSION MODELS (MDMS)

MDM's forward process gradually corrupts a clean sequence x_0 by independently replacing tokens with a dedicated mask token $\mathbf{m}$. The mask state is absorbing, i.e., the final state of the corruption process x_1 is an entirely masked sequence. To generate data, a backward process reconstructs the sequence by iteratively replacing mask tokens with clean ones. Clean tokens are estimated with a denoising model, typically a bidirectional transformer with learnable parameters θ, which maps the current partially masked state x_t to a predicted distribution over the vocabulary at each currently masked position $i \in \mathcal{M}_t$, denoted as $p_\theta(x_0^i \mid x_t)$. The denoising model is trained to match the true conditional data distribution $q(x_0^i \mid x_t)$ by minimizing a negative evidence lower bound.

2.3 THE CHALLENGE OF PARALLEL DECODING

Reversing the forward process during decoding would require *uniformly at random* selecting a subset $S \subseteq \mathcal{M}_t$ of masked positions and replacing it with a sample from the joint conditional distribution over clean sub-sequences x_0^S, i.e., $q(x_0^S \mid x_t)$. However, to avoid a combinatorial explosion of logits, MDMs approximate the true conditional joint probability as a product of independent marginals: $p_\theta(x_0 = c \mid x_t) = \prod_{i=1}^L p_\theta(x_0^i = c^i \mid x_t) =: \prod_i p_{\theta,t}^i(c)$. Here, $p_{\theta,t}^i(\cdot)$ denotes the marginal distribution at position i conditioned on x_t. Unfortunately, such a conditional independence bias results in an irreducible approximation error that equals the total conditional correlation of the sequence,

$C(x_0^S \mid x_t) := D_{\mathrm{KL}}\big(q(x_0^S \mid x_t) \,\|\, \prod_{j\in S} q(x_0^j \mid x_t)\big)$, as it holds that

$$\min_{p^j \in \Delta\ \forall j\in S} D_{\mathrm{KL}}(q(x_0^S \mid x_t) \,\big\|\, \prod\nolimits_{j\in S} p^j) = C(x_0^S \mid x_t).$$

Variants and corollaries of this statement were derived by Huang et al. (2022); Ben-Hamu et al. (2025); Liu et al. (2025); Kang et al. (2026); Zhong et al. (2026); Hersche et al. (2026). As a result, updating randomly drawn positions based on content selected entirely from isolated per-token marginals will inevitably result in the described approximation error.

To improve the generation quality, prior work aims to avoid committing sets S with high total conditional correlation. We categorize prior unmasking strategies into two families (see also Figure 1):

Marginal-only. These methods rely on the premise that positions with sharp, low-entropy marginal distributions can be safely committed in isolation (here the joint distribution approaches the product of marginals). They compute a scalar score per position (e.g., max probability or entropy) and select high-confidence positions based on a threshold, top-k cardinality cap, or cumulative budget.

Dependency-aware. These methods introduce a scalar $s^{i,j} \geq 0$, measuring how strongly positions i and j interact. They typically use $s^{i,j}$ to formulate a penalty or budget in a greedy or variational selection procedure. As a result, they select sets S with small total conditional correlation.

In both families, the value assignment fully relies on the sampled (or argmaxed) model prediction and is completely independent of the position selection. Hence, they answer *which positions* are safe to jointly commit, but not *which values*. Consequently, these approaches overlook the joint dependencies between positions and their assigned values, which ConGO aims to address.

3 ConGO: Consistency-guided unmasking

The existing value-blind rules cannot validate candidate values at commit time, affecting both throughput and accuracy: when faced with a coupled pair as in the example of Figure 1, they can only defer one position, spending an extra step on a pair whose values are in fact jointly determined, or commit both from the product of the marginals and risk an inconsistent pair. Two insights on an unmasking rule follow. First, dependence is not incompatibility: a coupled pair may still impose a unique consistent assignment, and a rule that picks positions and values jointly can commit the pair in one step rather than deferring. Independence is therefore sufficient for safe parallel commitment, but not necessary. Second, and more fundamentally, external knowledge is stated at the level of values (e.g., *these two cells may not both be* 4), so representing it requires couplings indexed by candidate values, not by positions. This work builds an unmasking objective around these two observations. Sec. 3.1 states it in general form, Sec. 3.2 instantiates it for different sources of value couplings, and Sec. 3.3 discusses its cost.

3.1 The ConGO unmasking objective

We keep the premise of position-coupled unmasking, i.e., that unmasking should account for interactions between positions, but let the unmasking rule choose values as well. To that end, we construct a shortlist $\mathcal{C}^i$ of candidate tokens for each position i and introduce a variable $z^{i,c} \in \{0,1\}$ for the decision to commit position i with value c. At each step, we select the assignment maximizing

$$\max_{z} \sum_{i\in\mathcal{M}_t} \sum_{c\in\mathcal{C}^i} h_t^{i,c}\, z^{i,c} + \sum_{\substack{i,j\in\mathcal{M}_t \\ i<j}} \sum_{\substack{c\in\mathcal{C}^i, \\ c'\in\mathcal{C}^j}} J^{i,j}(c,c')\, z^{i,c} z^{j,c'} \quad \text{s.t.} \quad \forall i \in \mathcal{M}_t : \sum_{c\in\mathcal{C}^i} z^{i,c} \leq 1. \quad (1)$$

Inequality constraint: ensures that a position either remains masked or is assigned a single value. Since an empty row in z defers updating i to a later step, the maximizer of Equation 1 decides both the number of committed tokens and their content. Folding the inequality constraint into the objective as a penalty, $-\mu \sum_i \sum_{c<c'\in\mathcal{C}^i} z^{i,c} z^{i,c'}$ with large $\mu > 0$, reduces Equation 1 to a quadratic unconstrained binary optimization (QUBO) problem. Other linear constraints, such as cardinality budgets or pairwise exclusions, can be folded into the objective likewise, or imposed explicitly when solving Equation 1 as a mixed-integer program.

Linear coefficient $h_t^{i,c}$: rewards committing position i at value c to ensure progress. Typically, it is informed by the denoising model, either directly through the marginal or through the score of an existing unmasking rule, and can include a consistency check with already-committed tokens.

Quadratic coefficient $J^{i,j}(c, c')$: scores the pair (c, c') as negative when the two values are incompatible, and as non-negative otherwise. Unlike dependency-aware samplers, which commit only near-independent sets, Equation 1 can commit a strongly coupled pair if it finds values that agree.

Remark. Setting $J^{i,j}(\cdot,\cdot) = 0$ reduces Equation 1 to a separable program over positions, and the marginal-only family is recovered by the choice of $h_t^{i,c}$. Confidence thresholding takes $h_t^{i,c} = p_{\theta,t}^i(c) - \beta$ for a threshold $\beta > 0$; top-k takes $h_t^{i,c} = p_{\theta,t}^i(c)$ subject to $\sum_{i\in\mathcal{M}_t}\sum_c z^{i,c} = k$; and the entropy-bound rule of Ben-Hamu et al. (2025) takes $h_t^{i,c} = 1$ subject to $\sum_{i\in\mathcal{M}_t} H(p_{\theta,t}^i)\sum_c z^{i,c} \leq \beta$, where $H(p_{\theta,t}^i)$ is the entropy of the marginal $p_{\theta,t}^i$. Each of these can be solved trivially by sorting, though Ivanov (2025) reports benefits from solving the associated QUBO objective instead; we attribute the benefits to the noise introduced by approximate solvers. Similarly, as shown in Sec. 3.2.3, dependency-aware methods can be obtained as a special case of Equation 1.

3.2 Instantiating the coupling

The coupling J in the objective of Equation 1 encodes which value pairs are consistent, and the reliability of this knowledge varies across domains. The extent of ConGO's intervention should therefore depend on this reliability. Exact external rules provide hard constraints that exclude incompatible value pairs, allowing ConGO to select both positions and values (Sec. 3.2.1). Learning the coupling from data yields soft constraints that reward jointly probable pairs and penalize jointly improbable ones. Such priors may be too weak to override the denoiser's choice of value, in which case J decides only which positions to commit, complementing either marginal-only (Sec. 3.2.2) or dependency-aware rules (Sec. 3.2.3).

3.2.1 Hard external constraints

Suppose a binary consistency check is available for every value pair, returning whether values c at i and c' at j can co-occur in a valid solution. We encode it as $J^{i,j}(c, c') = -\lambda$ if the assignment is inconsistent and 0 otherwise, and we set $h_t^{i,c} = 1$ so that every commitment has equal value, while the marginals enter only through the shortlist selection $\mathcal{C}^i$, from which values inconsistent with already-committed tokens are removed. For $\lambda > 0$ large enough that no violation can pay for itself, Equation 1 is exactly the maximum consistent subset problem: commit as many positions as possible, at values that are pairwise compatible and agree with the already-committed context. Note that shortlist sizes $|\mathcal{C}^i| > 1$ extend this beyond position filtering: at $|\mathcal{C}^i| = 1$ the program can only accept or reject a position based on the denoiser's $\arg\max$ or sampled value, whereas at $|\mathcal{C}^i| > 1$, it can commit a position at its secondary-choice value if it contributes to commitments elsewhere.

3.2.2 Learned consistency priors

When no hard external constraints are available, the coupling can instead be estimated from data. In natural language and code, pairwise dependencies are often strongest among nearby tokens. We therefore restrict J to adjacent positions and score a pair of candidate values by their pointwise mutual information (pmi) under a fixed bigram model b:

$$J^{i,i+1}(c, c') = \lambda \cdot \mathrm{pmi}(c; c'), \qquad \mathrm{pmi}(c; c') = \log b(c, c') - \log b(c) - \log b(c') \tag{2}$$

for $\lambda > 0, c \in \mathcal{C}^i$ and $c' \in \mathcal{C}^{i+1}$, with $J^{i,j} = 0$ for non-adjacent pairs.

As the bigram prior is too coarse to overrule the denoiser's choice of value, we keep a single candidate per position, $\mathcal{C}^i = \{\hat{c}^i\}$ with $\hat{c}^i = \arg\max_c p_{\theta,t}^i(c)$, so that a step only chooses the set $\mathcal{S} \subseteq \mathcal{M}_t$ of positions committed at these values. We note that considering only a single value per position is a design choice rather than our method's limitation; an analogous objective can still be formulated to optimize over different values per position. To combine the value-aware coupling with the denoiser's marginals in a principled way, we start from the goal of an unmasking step: committing as many positions as possible while keeping them jointly probable:

$$\max_{\mathcal{S}} \;\; c_0\,|\mathcal{S}| + \log q\big(x^{\mathcal{S}} \mid x_t\big)\,, \tag{3}$$

where c_0 rewards each additional commitment. This is intractable, since $q(x^{\mathcal{S}} \mid x_t)$ is the conditional joint the denoiser does not model. Under a first-order approximation, however, it reduces to an instance of Equation 1.

Proposition 1. *Suppose that conditioned on x_t, a committed value depends on the others committed in the same step only through its direct predecessor. When using conditional pointwise mutual information as a joint score $J^{i-1,i}(\hat{c}^{i-1}, \hat{c}^i | x_t)$, Equation 3 is equivalent to*

$$\max_{z \in \{0,1\}^{|\mathcal{M}_t|}} \sum_{i \in \mathcal{M}_t} h_t^i \, z^i + \sum_{i \,:\, i-1,\, i \in \mathcal{M}_t} J^{i-1,i}(\hat{c}^{i-1}, \hat{c}^i | x_t) \, z^{i-1} z^i, \tag{4}$$

for $z^i := z^{i,\hat{c}^i}$, $\mathcal{S} = \{i : z^i = 1\}$, $h_t^i = c_0 + \log p_{\theta,t}^i(\hat{c}^i)$, and $\lambda = 1$.

The proof in App. B.2 expands the chain rule so that, when the neighboring positions are jointly committed, the corresponding conditional splits into a marginal, contributing to a linear term, and a conditional pmi correction, contributing to a quadratic term. In practice, we use the objective from Equation 4, whereby we approximate the conditional pmi with a global, unconditional one from Equation 2. The bigram is only one instance of a learned prior: any signed, value-indexed estimate of pairwise consistency, e.g., a context-aware one accessed from the denoiser's hidden states as proposed in DFlash2 (Inco AI, 2026), plugs into J unchanged. Since the unconditional bigram models estimate these corrections coarsely, we leave λ as a hyperparameter to rescale the contribution of the bigram. Optionally, a candidate $\hat{c}^i$ can be scored against an already-committed predecessor x_t^{i-1} through an additional linear term $\mathrm{pmi}\big(x_t^{i-1}; \hat{c}^i\big)$ instead of only relying on the denoiser $p_{\theta,t}$; similarly, consistency against an already committed successor x_t^{i+1} can be scored. However, these additional consistency scores did not show any benefits in our experiments. Alg. 1 in App. B.1 provides a detailed description of one decoding step.

3.2.3 Learned consistency priors and dependency-aware rules

The previous sections used J to incorporate value-aware knowledge, either exact or learned. However, J can also encompass value-blind signals, such as the pairwise dependency estimates, and combine them with value-aware ones in a single objective. We demonstrate both steps on DAPD (Kim et al., 2026): we first express its value-blind dependency penalty as an instance of Equation 1, and then augment it with our value-aware bigram coupling of Equation 2.

DAPD reads a value-blind interaction score $s^{i,j}$ off the self-attention map and prevents co-commitment of two masked positions if $s^{i,j} > \tau_t$ for a threshold $\tau_t > 0$. Each position is scored by its confidence-weighted degree $h_t^i = \tilde{d}^i \cdot \mathrm{conf}_t^i$, where $\mathrm{conf}_t^i = \max_c p_{\theta,t}^i(c)$ and $\tilde{d}^i$ is the normalized attention degree $\sum_{j \neq i} s^{i,j}$. Positions are then checked in decreasing order of h_t^i and committed unless connected to a position already selected, in the style of the Welsh–Powell coloring heuristic. This selection is expressible in Equation 1: fixing a single candidate per position collapses the assignment variables to position indicators $z^i := z^{i,\hat{c}^i}$, the linear term is set to DAPD's score h_t^i, and the coupling is value-independent and hard, $J^{i,j} = -\infty \cdot \mathbb{I}[s^{i,j} > \tau_t]$. Alternatively, this rule can be imposed explicitly as $z^i + z^j \leq 1$, turning Equation 1 into a maximum-weight independent set problem, for which DAPD's greedy rule is the standard heuristic.

We augment DAPD by keeping its linear score and replacing its hard coupling between adjacent positions with the bigram coupling of Equation 2, setting $J^{i,j}(\hat{c}^i, \hat{c}^j)$ to $\lambda \cdot \mathrm{npmi}(\hat{c}^i; \hat{c}^j)$ if $j = i+1$, and to $-\infty \cdot \mathbb{I}[s^{i,j} > \tau_t]$ otherwise, where npmi is normalized pmi. Adjacent positions are thus no longer deferred merely because attention marks them as dependent. They are committed together when their values are jointly probable under the bigram prior, and penalized when they are not.

3.3 Computational complexity

While QUBO is NP-hard in general (Pardalos and Jha, 1992), the instances arising from Equation 1 are small and highly structured. In commonly used semi-AR block diffusion, the number of variables, $\sum_{i \in \mathcal{M}_t} |\mathcal{C}^i|$, is bounded by the decoding block size of typically 32–256 tokens (e.g., Nie et al., 2025; DiffusionGemma Team et al., 2026), and can be reduced further with small shortlists $\mathcal{C}^i$ or restriction of $\mathcal{M}_t$ to fewer positions. The couplings are sparse as well: the bigram instantiation couples only neighboring positions, whereas the Sudoku constraints couple each cell only to those

Table 1: Exact solve rate (%) on Sudoku for varying NFEs on easy (in-distribution) and hard (top decile difficulty out-of-distribution) test set. $|\mathcal{C}^i|$ is the number of candidate values per position.

	Easy			Hard		
	Accuracy (%) @NFE			Accuracy (%) @NFE		
Method	5	9	21	5	9	21
Random order	0.5	1.4	2.7	0.1	0.4	1.0
Max probability	28.4	64.9	85.4	4.3	14.1	26.4
Probability margin	28.5	65.2	<u>85.5</u>	3.8	12.8	24.9
Negative entropy	23.9	62.1	84.5	3.2	13.2	<u>26.8</u>
Greedy	<u>29.8</u>	<u>65.4</u>	84.8	<u>4.8</u>	<u>14.6</u>	25.8
ConGO, $\lvert\mathcal{C}^i\rvert$=1	31.4	66.4	85.2	5.4	15.4	26.6
ConGO, $\lvert\mathcal{C}^i\rvert$=2	54.6	79.6	90.8	13.9	25.3	35.7
ConGO, $\lvert\mathcal{C}^i\rvert$=3	**81.4**	**93.3**	**97.6**	**45.3**	**59.8**	**69.2**

sharing its row, column, or box. Dedicated branch-and-cut solvers are efficient in both regimes, as the timings provided for Gurobi solver (Gurobi Optimization, LLC, 2026) in Sec. 4 confirm.

4 Experiments

We evaluate ConGO in two scenarios: (1) 9×9 Sudoku puzzle solving, where exact consistency is known for any pair of cells, allowing for joint value and position selection; and (2) code generation, where a static bigram prior models the coupling, which enhances both the marginal-only and dependency-aware position selection.

4.1 Hard external constraints for sudoku

MDM. We train a 5.4M-parameter bidirectional transformer to solve Sudoku puzzles with a standard MDM recipe, used for example by Kim et al. (2025a); Mohamud et al. (2026). Following Kim et al. (2025a), we report the percentage of exact solves on 100k held-out puzzles from the dataset of Shah et al. (2024) and, to probe out-of-distribution (OOD) behavior, on the hardest decile of the complementary *hard* split. At each step, we unmask the top-k positions under one of five scoring rules: random, maximum marginal probability (Zheng et al., 2024), probability margin (Kim et al., 2025a), negative entropy (Ben-Hamu et al., 2025), and a self-created greedy constraint-aware rule. App. C.1 provides details on the architecture, baselines, training, and inference.

In distribution, ConGO with $|\mathcal{C}^i| \geq 2$ outperforms all baselines at all NFEs, defining the accuracy-NFE Pareto front. When each cell contributes a single candidate ($|\mathcal{C}^i|$=1), ConGO can only accept or reject the denoiser's $\arg\max$, consequently achieving comparable results to the baseline scoring heuristics. With $|\mathcal{C}^i| > 1$, it can instead take a cell's secondary-choice digits when doing so admits further commitments elsewhere. Hence, the gain stems from choosing among candidates rather than from purely filtering positions based on the known constraints. The advantage grows in OOD, where every method degrades, as the best baseline falls from 85.5% to 24.9%. However, ConGO's margin against the baseline widens from 12.1 pp to 42.4 pp. Additional results on sequential decoding and generation times are provided in App. D.

Generalization to UDM. We also evaluate our method in combination with a UDM, namely DiffusionGemma (DiffusionGemma Team et al., 2026), showing matched accuracy at 8% speed-up, and a 4.5 pp accuracy increase with 11% fewer NFEs (see App. D).

4.2 Learned constraints for code generation

We evaluate on code generation, where an unconditional bigram can be trained to model the syntax. We test LLaDA-8B-Instruct (Nie et al., 2025) on HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021b). All methods share a generation length of 256, a block length of 64, and a maximum budget of 256 denoising steps. We compare against confidence-ranked sequential decoding

Table 2: Accuracy (%), average NFE, and average wall-clock time per sequence (seconds on an NVIDIA H100 GPU) on HumanEval and MBPP with LLaDA-8B-Instruct.

	HumanEval			MBPP		
	Acc (%) ↑	NFE ↓	Time (s) ↓	Acc (%) ↑	NFE ↓	Time (s) ↓
Top-1	37.2	256	5.8	18.6	256	9.7
EB	39.0	111	2.4	20.0	53	2.2
KLASS	37.2	151	3.4	24.6	54	2.4
Fast dLLM (0.9)	36.6	92	1.9	20.2	41	1.6
Fast dLLM (0.7)	37.2	52	1.1	23.4	22	**0.9**
DAPD	**42.1**	49	1.1	**35.2**	28	1.3
ConGO-PMI	36.6	**40**	**1.0**	30.0	**22**	1.0
ConGO-PMI-DAPD	42.0	42	1.4	35.0	23	1.3

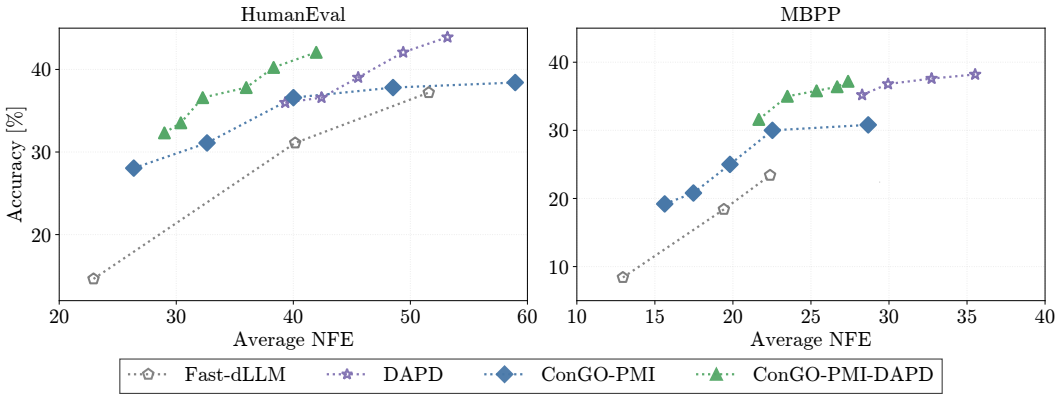

Figure 2: Accuracy (%) vs. average NFE of LLaDA-8B-Instruct on HumanEval and MBPP. Lines show the per-method Pareto fronts obtained by sweeping the confidence threshold (Fast-dLLM, ConGO-PMI) or the dependency threshold (DAPD, ConGO-PMI-DAPD).

(Top-1), EB (Ben-Hamu et al., 2025), Fast-dLLM (Wu et al., 2026), KLASS (Kim et al., 2025b), and DAPD (Kim et al., 2026), each configured with their recommended hyperparameters borrowed from Kim et al. (2026). Besides its independent-set selection, DAPD commits confident tokens through a separate, marginal rule that comes in two variants: DAPD-direct commits every token with confidence 1, and DAPD-staged additionally commits every token above 0.7 once half of the sequence is unmasked. We evaluate the variants of Secs. 3.2.2 and 3.2.3, denoted ConGO-PMI and ConGO-PMI-DAPD, with the same bigram model. ConGO-PMI-DAPD builds on DAPD-direct: instead of relying on a value-blind threshold, it uses the value-aware coupling to co-commit consistent tokens. App. C.2 details the protocol, bigram estimation, hardware, and hyperparameters.

Figure 2 shows that both variants push the Pareto front of their base rule toward fewer NFEs, and Table 2 lists selected points (for Fast-dLLM, the default threshold 0.9 and 0.7, its most accurate setting). At matched accuracy, ConGO-PMI requires 23% fewer NFEs than Fast-dLLM on HumanEval, and ConGO-PMI-DAPD requires 14% and 18% fewer than DAPD on HumanEval and MBPP, respectively. At matched NFE on MBPP, ConGO-PMI improves accuracy over thresholding by a relative gain of 28%. The gains are largest at low NFE, where aggressive thresholds co-commit many adjacent tokens and value-aware coupling matters most. For ConGO-PMI, the NFE savings carry over to wall-clock time. For ConGO-PMI-DAPD, the solver overhead offsets them on HumanEval, which motivates reducing the solving cost further. The complete analysis, including dominated configurations and the DAPD-staged baseline, are shown in Figure 3 in App. D.

5 Related work

Marginal-only selection. In these strategies, each masked position is scored based on its own predictive distribution before a separable selection operator is applied: top-k confidence or entropy (Nie et al., 2025; Ye et al., 2025), a global confidence threshold (Wu et al., 2026), an entropy budget (Ben-Hamu et al., 2025), between-step KL stability (Kim et al., 2025b), relaxed commitment near high-confidence anchors (Kong et al., 2025). Ni et al. (2026) find that delaying the decoding of low-confidence tokens (e.g., branching tokens with high entropy) can actually be detrimental to reasoning performance. One potential solution is to simulate the outcome of different update sets in order to maximize a model-free reward (e.g., maximum information gain as reflected by average entropy reduction) (Yang et al., 2026; Lee et al., 2026; Leang et al., 2026); however, this requires multiple denoising forward passes, which can be limiting in concurrent servings. Finally, all such marginal rules can be recovered within our ConGO framework by removing the quadratic term and integrating the corresponding linear constraint and the marginal score as a linear term in Equation 1.

Dependency-aware selection. More recent works estimate interactions between positions to jointly commit tokens with low dependency. DAPD (Kim et al., 2026) uses pairwise self-attention scores to build a thresholded dependency graph and selects a maximum independent set by a Welsh–Powell-style coloring heuristic; DAWN (Luo et al., 2026) builds a similar conflict graph while correcting attention-sink artifacts, and CLAD (Qi et al., 2026) lifts the decision from tokens to confidence-induced spans. DEMASK (Ringel et al., 2026) trains a predictor on the denoiser's final hidden states to estimate pairwise conditional influence and greedily selects positions under a cumulative dependency bound. ADAS (Sahin et al., 2026) keeps attention continuous and uses it as a soft marginal penalty to greedily discount a candidate that attends strongly to already selected positions. In similar spirit, structured mean-field commit rules (Zoabi et al., 2026) apply a variational correction; Zhou et al. (2026) aims to commit tokens with high attention to already committed tokens; and Luxembourg et al. (2026) aim to reduce token interaction via spatially dilated grouping. These methods aim to find *independent* positions to safely commit. In contrast, our permissive ConGO embraces *consistent* pairs, allowing highly correlated tokens to be safely committed together.

Constrained decoding and optimization. Constrained decoding has been adapted to MDMs, covering regular constraints (Suresh et al., 2025) and context-free grammars (Mündler et al., 2026; Jin and Han, 2026); however, validating the grammar poses a bottleneck. Guided generation adapts the marginals based on an external reward (Dou et al., 2026; Shao et al., 2026). Similarly, CDD (Cardei et al., 2025) iteratively adapts the marginals using a projection operator, nudging the distribution towards constraint-satisfying solutions. Besides adapting the marginals, recent work explores combinatorial optimization for update selection. For instance, Ivanov (2025) frame non-sequential text synthesis as a QUBO problem, but their formulation lacks value-awareness required to effectively couple constrained token-values, reducing to a top-k sorting of marginal confidences. In contrast, our ConGO framework explicitly indexes couplings by candidate values, allowing the optimization to balance marginal confidence with both hard symbolic constraints and learned priors.

App. A covers more related works on learned policies, denoiser distribution improvements, and augmented estimation and sampling from the conditional joint.

6 Conclusion

We recast the update step of discrete diffusion as a joint decision over positions and values. Unlike value-blind prior samplers, which have to defer coupled positions even when compatible values exist, ConGO can consider multiple values at each position and solve for set selection and assignment, subsuming prior rules as special cases while remaining tractable. Exact constraints buy up to 42.4 pp on out-of-distribution Sudoku, while a bigram prior, integrated into both marginal-only and dependency-aware decoding, improves their accuracy–NFE Pareto front on LLaDA-8B-Instruct.

Limitations and future work. The solver cost is non-negligible, and end-to-end latency benefits primarily emerge in larger DDMs (LLaDA-8B and DiffusionGemma). We see future work in reducing the solving time, e.g., through problem reduction via position pre-filtering. Moreover, the consistency models are deliberately simple; richer learned priors are the natural next step.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing methods, specifically for writing and editing code for the proposed decoding procedure. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide ingredients for or write proofs, propose or refine hypotheses, design research methodology or experiments, clean or reformat datasets, or interpret results. Assisting with translation and supporting qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to edit the manuscript for readability, including rephrasing sentences, improving transitions, and polishing the structure and wording of individual sections. We have reviewed all AI-assisted work: edited text was checked against our own derivations and results to ensure that no technical content, claims, or citations were altered or introduced, and AI-assisted code was reviewed and verified for correctness before being used to produce the reported results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have made several efforts to ensure the reproducibility of our results. The general ConGO objective and all its instantiations are fully specified in Sec. 3, including the exact choices of the linear and quadratic coefficients for each setting provided in App. C.1 and C.2. Alg. 1 in App. B.1 provides pseudocode for one ConGO decoding step, and App. B.2 states the assumptions and gives a complete proof of Proposition 1. App. C provides full implementation, training, and evaluation details for all models and experiments, including architectures, datasets, hyperparameters, and the exact optimization problems solved at each decoding step. All datasets and benchmarks we use are publicly available, and all optimization problems are solved with the Gurobi solver, which is freely available under an academic license. As indicated by ICLR's author guidelines, we are available to share the code anonymously during the rebuttal period.

REFERENCES

Hikaru Asano, Tadashi Kozuno, Kuniaki Saito, and Yukino Baba. Where-to-Unmask: Ground-Truth-Guided Unmasking Order Learning for Masked Diffusion Language Models. *arXiv preprint arXiv:2602.09501*, February 2026. doi: 10.48550/arXiv.2602.09501. URL http://arxiv.org/abs/2602.09501.

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 (NeurIPS)*, volume 34, November 2021a. URL https://openreview.net/forum?id=h7-XixPCAL.

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. URL https://arxiv.org/abs/2108.07732.

Haolei Bai, Lingcheng Kong, Xueyi Chen, Jianmian Wang, Zhiqiang Tao, and Huan Wang. DICE: Diffusion Large Language Models Excel at Generating CUDA Kernels. *arXiv preprint arXiv:2602.11715*, June 2026. doi: 10.48550/arXiv.2602.11715. URL http://arxiv.org/abs/2602.11715.

Parikshit Bansal and Sujay Sanghavi. Enabling Approximate Joint Sampling in Diffusion LMs. *arXiv preprint arXiv:2509.22738*, September 2025. doi: 10.48550/arXiv.2509.22738. URL http://arxiv.org/abs/2509.22738.

Wenrui Bao, Zhiben Chen, Dan Xu, and Yuzhang Shang. Learning to Parallel: Accelerating Diffusion Large Language Models via Learnable Parallel Decoding. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=bFJ8Sdr224.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas S Nolte, and Brian Karrer. Accelerated Sampling from Masked Diffusion Models via Entropy Bounded Unmasking. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. doi: 10.52202/085713-1874. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/510e22f4a2da5212a64f1591736e2eaf-Abstract-Conference.html.

Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, Zhuocheng Gong, Yanmei Gu, Jian Guan, Kaiyuan Guan, Hongliang He, Zenan Huang, Juyong Jiang, Zhonghui Jiang, Zhenzhong Lan, Chengxi Li, Jianguo Li, Zehuan Li, Huabin Liu, Lin Liu, Guoshan Lu, Yuan Lu, Yuxin Ma, Xingyu Mou, Zhenxuan Pan, Kaida Qiu, Yuji Ren, Jianfeng Tan, Yiding Tian, Zian Wang, Lanning Wei, Tao Wu, Yipeng Xing, Wentao Ye, Liangyu Zha, Tianze Zhang, Xiaolu Zhang, Junbo Zhao, Da Zheng, Hao Zhong, Wanli Zhong, Jun Zhou, Junlin Zhou, Liwang Zhu, Muzhi Zhu, and Yihong Zhuang. LLaDA2.1: Speeding Up Text Diffusion via Token Editing. *arXiv preprint arxiv:2602.08676*, February 2026. doi: 10.48550/arXiv.2602.08676. URL http://arxiv.org/abs/2602.08676.

Andrew Campbell, Joe Benton, and Valentin De Bortoli. A Continuous Time Framework for Discrete Denoising Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. URL https://openreview.net/pdf?id=DmT862YAieY.

Michael Cardei, Jacob K. Christopher, Bhavya Kailkhura, Tom Hartvigsen, and Ferdinando Fioretto. Constrained Discrete Diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pages 12383–12416, 2025. URL https://openreview.net/forum?id=Es4s9dtCjR.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked Generative Image Transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11305–11315, June 2022. doi: 10.1109/CVPR52688.2022.01103. URL https://ieeexplore.ieee.org/document/9878676. ISSN: 2575-7075.

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. *arXiv preprint arXiv:2107.03374*, July 2021. doi: 10.48550/arXiv.2107.03374. URL http://arxiv.org/abs/2107.03374.

Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. dParallel: Learnable Parallel Decoding for dLLMs. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=hVOcstAURb.

DiffusionGemma Team, Adrien Ali Taïga, James Assiene, Daniele Calandriello, Rahma Chaabouni, João Gante, Tamara von Glehn, Nate Keating, Chris Knutsen, Martin Kukla, Tianlin Liu, Ivan Lobov, Ofir Nabati, João Gabriel Oliveira, Nicolas Perez-Nieves, Nastasia Prutianova, Bobak Shahriari, Jean Tarbouriech, Pavel Tyletski, Çağlar Ünlü, Cindy Wu, Glenn Cameron, Jerome Connor, Sertan Girgin, Maarten Grootendorst, Alon Levkovitch, Eliya Nachmani, Omar Sanseviero, Piotr Stanczyk, Quentin Berthet, Andrew Campbell, Clément Crepy, Valentin De Bortoli, Arnaud Doucet, Romuald Elie, Alexandre Galashov, Klaus Greff, Alexis Jacq, David Ruhe, Yu-Han Wu, Sebastian Flennerhag, Brendan O'Donoghue, George Scrivener, and Shantanu Thakoor. DiffusionGemma Technical Report. *arXiv preprint arXiv:2608.00146*, July 2026. doi: 10.48550/arXiv.2608.00146. URL http://arxiv.org/abs/2608.00146.

Hongkun Dou, Zike Chen, Fengji Li, Hongjue Li, and Yue Deng. Plug-and-Play Guidance for Discrete Diffusion Models via Gradient-Informed Logit Correction. In *Forty-third International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=IXsOS3hF8Y.

Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical Benefit and Limitation of Diffusion Language Model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=fGBCRZQVse.

Shengyu Feng, Tarun Suresh, and Yiming Yang. Unsupervised Diffusion Solver for Combinatorial Optimization via Combinatorial Adjoint Matching. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=y6LGSCnNOz.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2026. URL https://www.gurobi.com.

Michael Hersche, Nicolas Menet, Ronan Tanios, and Abbas Rahimi. Locally Coherent Parallel Decoding in Diffusion Language Models. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=hn4ndkaD6x.

Chunsan Hong, Seonho An, Min-Soo Kim, and Jong Chul Ye. Improving Discrete Diffusion Unmasking Policies Beyond Explicit Reference Policies. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=on6cb46OhD.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Fei Huang, Tianhua Tao, Hao Zhou, Lei Li, and Minlie Huang. On the Learning of Non-Autoregressive Transformers. In *International Conference on Machine Learning (ICML)*, 2022. URL https://proceedings.mlr.press/v162/huang22k.html.

Inco AI. DFlash 2: Keep Drafting Parallel, August 2026. URL https://inco.ai/blog/dflash2/#a-lightweight-path-selector. Accessed: 2026-09-09.

Daniel Mingyi Israel, Guy Van den Broeck, and Aditya Grover. Accelerating Diffusion LLMs via Adaptive Parallel Decoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=xwqTt26NJf.

Adams Ivanov. A Paradigm Shift in Large Language Generation: Quantum-Inspired Diffusion for Non-Sequential Text Synthesis. 2025. URL https://www.academia.edu/128316764/A_Paradigm_Shift_in_Large_Language_Generation_Quantum_Inspired_Diffusion_for_Non_Sequential_Text_Synthesis.

Metod Jazbec, Theo X. Olausson, Louis Béthune, Pierre Ablin, Michael Kirchhof, João Monteiro, Victor Turrisi, Jason Ramapuram, and Marco Cuturi. Learning Unmasking Policies for Diffusion Language Models. In *Forty-third International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=F9NDKf5oPy&noteId=ISiDwTAmZP.

Hyundong Jin and Yo-Sub Han. EPIC: Efficient and Parallel Inference under CFG Constraints for Diffusion Language Models. *arXiv preprint arXiv:2606.00722*, 2026. doi: 10.48550/arxiv.2606.00722. URL https://arxiv.org/abs/2606.00722.

Ziqi Jin, Bin Wang, Xiang Lin, Lidong Bing, and Aixin Sun. On the Role of Discreteness in Diffusion LLMs. *arXiv preprint arXiv:2512.22630*, December 2025. doi: 10.48550/arXiv.2512.22630. URL http://arxiv.org/abs/2512.22630.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjae Lee, Yuchen Zeng, Shuibai Zhang, Coleman Hooper, Yuezhou Hu, Hyung Il Koo, Nam Ik Cho, and Kangwook Lee. ParallelBench: Understanding the Trade-offs of Parallel Decoding in Diffusion LLMs. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=OsZr5T7Cd0.

BumJun Kim, Dongjae Jeon, Moongyu Jeon, and Albert No. DAPD: Dependency-Aware Parallel Decoding via Attention for Diffusion LLMs. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=LZro8iqjiA.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the Worst, Plan for the Best: Understanding Token Ordering in Masked Diffusions. In *Forty-second International Conference on Machine Learning (ICML)*, June 2025a. URL https://openreview.net/forum?id=DjJmre5IkP.

Seo Hyun Kim, Sunwoo Hong, Hojung Jung, Youngrok Park, and Se-Young Yun. KLASS: KL-Guided Fast Inference in Masked Diffusion Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b. URL https://openreview.net/forum?id=gOG9Zoyn4R&noteId=Z84saXY69z.

Fanheng Kong, Jingyuan Zhang, Yahui Liu, Zirui Wu, Yu Tian, Victoria W, and Guorui Zhou. Accelerating Diffusion LLM Inference via Local Determinism Propagation. *arXiv preprint arXiv:2510.07081*, October 2025. doi: 10.48550/arXiv.2510.07081. URL http://arxiv.org/abs/2510.07081.

Joshua Ong Jun Leang, Yu Zhao, Mihaela C. Stoian, Wenda Li, Shay B. Cohen, and Eleonora Giunchiglia. Can I Have Your Order? Monte-Carlo Tree Search for Slot Filling Ordering in Diffusion Language Models. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=mEfh2cH3AW.

Sanghyun Lee, Seungryong Kim, Jongho Park, and Dongmin Park. Lookahead Unmasking Elicits Reliable Decoding in Diffusion Language Models. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=OdM5JRVEiK.

Ian Li, Zilei Shao, Benjie Wang, Rose Yu, Guy Van den Broeck, and Anji Liu. Breaking the Factorization Barrier in Diffusion Language Models. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=euv8IoKFbh.

Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete Copula Diffusion. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=FXw0okNcOb.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning (ICML)*, volume 235, July 2024. URL https://openreview.net/forum?id=CNicRIVIPA.

Lizhuo Luo, Zhuoran Shi, Jiajun Luo, Zhi Wang, Shen Ren, Wenya Wang, and Tianwei Zhang. DAWN: Dependency-Aware Fast Inference for Diffusion LLMs. *arXiv preprint arXiv:2602.06953*, February 2026. doi: 10.48550/arXiv.2602.06953. URL http://arxiv.org/abs/2602.06953.

Omer Luxembourg, Haim H. Permuter, and Eliya Nachmani. Plan for Speed: Dilated Scheduling for Masked Diffusion Language Models. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=DbzAsMRNhy.

Jama Hussein Mohamud, Mohsin Hasan, Mirco Ravanelli, and Yoshua Bengio. Adaptive Order Policies for Masked Diffusion. *arXiv preprint arXiv:2606.00295*, May 2026. doi: 10.48550/arXiv.2606.00295. URL http://arxiv.org/abs/2606.00295.

Niels Mündler, Jasper Dekoninck, and Martin Vechev. Constrained Decoding of Diffusion LLMs with Context-Free Grammars. In *The Fourteenth International Conference on Learning Representations (ICLR)*, June 2026. URL https://openreview.net/forum?id=7Sph4KyeYO.

Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, and Gao Huang. The Flexibility Trap: Rethinking the Value of Arbitrary Order in Diffusion Language Models. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=kpgURPRMGf.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=KnqiC0znVF.

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 *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

Panos M Pardalos and Somesh Jha. Complexity of uniqueness and local search in quadratic 0–1 programming. *Operations research letters*, 11(2):119–123, 1992.

Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. Diffusion Beats Autoregressive in Data-Constrained Settings. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=W5Ht05jF4c.

Heqiang Qi, Wei Huang, Mingyuan Bai, and Xiangming Meng. Cluster-Level Attention-Guided Parallel Decoding for Masked Diffusion Language Models. *arXiv preprint arXiv:2605.29607*, May 2026. doi: 10.48550/arXiv.2605.29607. URL http://arxiv.org/abs/2605.29607.

David G. Radcliffe. 3 million sudoku puzzles with ratings, 2020. URL https://www.kaggle.com/dsv/1495975.

Yuji Ren, Chenkai Xu, Zhuocheng Gong, Jianguo Li, and Zhijie Deng. CForce: Boosting Parallel Decoding for dLLMs via Consistency Forcing. *arXiv preprint arXiv:2608.13925*, August 2026. doi: 10.48550/arXiv.2608.13925. URL http://arxiv.org/abs/2608.13925.

Liran Ringel, Ameen Ali, and Yaniv Romano. Dependency-Guided Parallel Decoding in Discrete Diffusion Language Models. *arXiv preprint:2604.02560*, April 2026. doi: 10.48550/arXiv.2604.02560. URL http://arxiv.org/abs/2604.02560.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized Interpolating Discrete Diffusion. In *Forty-second International Conference on Machine Learning (ICML)*, 2025. URL https://openreview.net/forum?id=rvZv7sDPV9.

Dimitri von Rütte, Janis Fluri, Omead Pooladzandi, Bernhard Schölkopf, Thomas Hofmann, and Antonio Orvieto. Scaling Behavior of Discrete Diffusion Language Models. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=GDYaNzxt9T.

Yusuf Sahin, Ahmed Rockey Saikia, Volkan Cevher, and Paolo Favaro. Attention-Discounted Adaptive Sampler for Masked Diffusion Language Models. *arXiv preprint arXiv:2606.10829*, July 2026. doi: 10.48550/arXiv.2606.10829. URL http://arxiv.org/abs/2606.10829.

Subham Sekhar Sahoo, Marianne Arriola, and Yair Schiff. Simple and Effective Masked Diffusion Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

Subham Sekhar Sahoo, Jean-Marie Lemercier, Zhihan Yang, Justin Deschenaux, Jingyu Liu, John Thickstun, and Ante Jukic. Scaling Beyond Masked Diffusion Language Models. In *The Forty-Third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=xKJG1CNtQM.

Kulin Shah, Nishanth Dikkala, Xin Wang, and Rina Panigrahy. Causal language modeling can elicit search and reasoning capabilities on logic puzzles. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 56674–56702, 2024. URL https://openreview.net/forum?id=i5PoejmWoC.

Lize Shao, Michael Cardei, Zichen Xie, Ferdinando Fioretto, and Wenxi Wang. Constrained Code Generation with Discrete Diffusion. *arXiv preprint arXiv:2605.16829*, May 2026. doi: 10.48550/arXiv.2605.16829. URL http://arxiv.org/abs/2605.16829.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. Simplified and Generalized Masked Diffusion for Discrete Data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://openreview.net/forum?id=xcqSOfHt4g.

Juntong Shi, Brian L. Trippe, Jure Leskovec, Stefano Ermon, and Minkai Xu. Diffusion Language Model Parallel Decoding via Product-of-Experts Bridge. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=iA7hUqvsvz.

Haocheng Sun, Cynthia Xin Wen, and Edward Hong Wang. Why mask diffusion does not work. *arXiv preprint arXiv:2510.03289*, September 2025. doi: 10.48550/arXiv.2510.03289. URL http://arxiv.org/abs/2510.03289.

Zipeng Sun, Can Chen, Ye Yuan, Haolun Wu, Jiayao Gu, Christopher Pal, and Xue Liu. Training Diffusion Language Models for Black-Box Optimization. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=Z7wI0sor6i.

Tarun Suresh, Debangshu Banerjee, Shubham Ugare, Sasa Misailovic, and Gagandeep Singh. DINGO: Constrained Inference for Diffusion LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=KaYMGsnZ4R.

Linxuan Wang, Ziyi Wang, Yikun Bai, Wei Deng, Guang Lin, and Qifan Song. Generalized Discrete Diffusion with Self-Correction. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=zQKlzKB6I9.

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 *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=3Z3Is6hnOT.

Jiawei Xu, Minghui Liu, Aakriti Agrawal, Yifan Chen, and Furong Huang. Scheduling Thoughts: Learning the Order of Thought in Diffusion Language Models. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=j4UhHaxCpq.

Kaisen Yang, Jayden Teoh, Kaicheng Yang, Yitong Zhang, and Alex Lamb. Improving Sampling for Masked Diffusion Models via Information Gain. In *Forty-third International Conference on Machine Learning (ICML)*, June 2026. URL https://openreview.net/forum?id=1HPTFpwktA.

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*, August 2025. doi: 10.48550/arXiv.2508.15487. URL http://arxiv.org/abs/2508.15487.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A Reparameterized Discrete Diffusion Model for Text Generation. In *First Conference on Language Modeling (COLM)*, August 2024. URL https://openreview.net/forum?id=PEQFHRUFca.

Yangyang Zhong, Yanmei Gu, Zhengqing Zang, Xiaomeng Li, Yuqi Ding, Xibei Jia, Yuting Shen, Zhenzhong Lan, Liwang Zhu, Weiping Liu, Junlin Zhou, Haisheng Liu, Zhong Xin Yu, Pengxin Luo, Donglian Qi, Yunfeng Yan, and Junbo Zhao. Parallelism and Generation Order in Masked Diffusion Language Models: Limits Today, Potential Tomorrow. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 7178–7218, San Diego, California, United States, January 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.357. URL https://aclanthology.org/2026.findings-acl.357/.

Xueyu Zhou, Yangrong Hu, and Jian Huang. DOS: Dependency-Oriented Sampler for Masked Diffusion Language Models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 17404–17419, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.861. URL https://aclanthology.org/2026.findings-acl.861/.

Tamim Zoabi, Ameen Ali, Liran Ringel, and Lior Wolf. Mean-Field Parallel Decoding for Discrete Diffusion Language Models. *arXiv preprint arXiv:2606.15805*, June 2026. doi: 10.48550/arXiv.2606.15805. URL http://arxiv.org/abs/2606.15805.

A Additional related work

Learned policies. Beyond dependency minimization, other methods directly learn an unmasking policy based on marginal-only (Xu et al., 2026; Bao et al., 2026) or dependency-aware (Asano et al., 2026; Hong et al., 2026; Jazbec et al., 2026) features. While effective, these policies are value-blind and isolate the content and position selection. Nevertheless, our method could consume their per-position policy scores (in h) and integrate pairwise, value-aware constraints or priors (in J).

Denoiser distribution improvements. An alternative direction aims to directly shape the denoiser's output distribution to enable more aggressive parallel decoding. dParallel (Chen et al., 2026) aims to sharpen the conditional marginals through certainty-forcing. CForce (Ren et al., 2026) aims to improve early predictions through distillation. Our ConGO could readily benefit from such improved denoisers, further improving the parallel decoding.

Augmented estimation and sampling from the conditional joint. APD (Israel et al., 2025) and PoE (Shi et al., 2026) use an AR model alongside the denoiser to approximate joint effects fully left-to-right; ADJUST Bansal and Sanghavi (2025) augments the MDM with a single-layer sequential MDM sampler; CoDiLA (Hersche et al., 2026) generates locally coherent blocks with an AR that is soft-conditioned on the MDM's marginal distribution; similarly, discrete copula diffusion Liu et al. (2025) combines MDM marginals with AR distributions but requires two MDM passes per update; and CoDD (Li et al., 2026) uses probabilistic circuits to estimate the conditional joint. Our ConGO can be combined with these methods, further allowing us to incorporate expert domain knowledge or learned priors to do value-aware updates.

B ConGO for code generation

This appendix provides more details on the pseudocode for one ConGO decoding step (App. B.1) and the proof of Proposition 1 (App. B.2).

B.1 Pseudoalgorithm

Alg. 1 demonstrates a single unmasking step with ConGO, when a bigram model is available. Line 2 limits the problem size by restricting the considered positions to w masked positions. The candidate positions can be selected based on different criteria, e.g., by considering only positions i with $\arg\max_c p^i_{\theta,t}(c) > \beta$ for some threshold $\beta > 0$, or based on a local window of neighboring tokens. Line 3 restricts the number of shortlisted candidates $|\mathcal{C}^i|$ per position to k, in case the optimization is also done over committed values. Line 4 assembles the linear coefficients from the parallelism bonus parameter c_0, the marginal, and the consistency with already-committed adjacent tokens, while pairwise interactions among candidates decided in the same step enter through J. The optimization problem 4 is then solved (line 5) and its solution is read as a set of positions and their values (line 6), an all-zero row meaning that the position is deferred. Since $z = 0$ is always feasible, the solution of the optimization problem can be empty, in which case the fallback in line 7 then commits the single best position-value pair, limiting the total number of steps not to exceed the sequence length.

Algorithm 1 ConGO unmasking step with bigram model

Require: masked state x_t, denoiser f_θ, coupling J, parallelism bonus c_0, shortlist size k, window size w

1: $p_{\theta,t} \leftarrow f_\theta(x_t)$ ▷ single forward pass
2: $\mathcal{M}_w \leftarrow$ a subset of w positions of $\mathcal{M}_t$ ▷ restrict candidate positions (optional)
3: $\mathcal{C}^i \leftarrow \text{top-}k\big(p^i_{\theta,t}\big)$ for $i \in \mathcal{M}_w$ ▷ restrict candidate values (optional)
4: $h_t^{i,c} \leftarrow c_0 + \log p^i_{\theta,t}(c) + \gamma\left(\left(\mathbb{I}[i-1 \notin \mathcal{M}_t]\,\mathrm{pmi}\big(x_t^{i-1} | c\big) + \mathbb{I}[i+1 \notin \mathcal{M}_t]\,\mathrm{pmi}\big(c; x_t^{i+1}\big)\right)\right)$
5: $z^\star \leftarrow$ solve 4 over $\mathcal{M}_w$
6: $S \leftarrow \{i : \exists c,\ z^\star_{ic} = 1\}$; $\quad v_i \leftarrow c$ where $z^\star_{ic} = 1$
7: **if** $S = \emptyset$ **then** commit $\arg\max_{i,c} p^i_{\theta,t}(c)$ ▷ guarantees progress
8: **return** x with $x^S \leftarrow v_S$

B.2 Proof of Proposition 1

For the unmasking step at time t, let the committed indices be $\mathcal{S} = \{i_1 < i_2 < \cdots < i_d\} \subseteq \mathcal{M}_t$. Throughout, the committed value at a position i is that of the single candidate per position, $x^i = \hat{c}^i$; the multi-candidate case is treated in the remark at the end.

Proof. By the chain rule along the increasing order,

$$\log q\Big(x^{\mathcal{S}} \mid x_t\Big) = \sum_{j=1}^{d} \log q\Big(x^{i_j} \mid x^{i_1}, \ldots, x^{i_{j-1}}, x_t\Big). \tag{5}$$

We now assume first-order dependence among simultaneous commitments, i.e., for every j, we approximate $q\big(x^{i_j} \mid x^{i_1}, \ldots, x^{i_{j-1}}, x_t\big)$ with $q\big(x^{i_j} \mid x^{i_j-1}, x_t\big)$ if $i_j - 1 \in \mathcal{S}$ and $q\big(x^{i_j} \mid x_t\big)$ otherwise. Applying this approximation, and then adding and subtracting $\log q(x^{i_j} \mid x_t)$ in each term whose predecessor is in $\mathcal{S}$, Equation 5 becomes

$$\begin{aligned}
\log q\Big(x^{\mathcal{S}} \mid x_t\Big) &= \sum_{j : i_j - 1 \notin \mathcal{S}} \log q\Big(x^{i_j} \mid x_t\Big) + \sum_{j\,:\,i_j - 1 \in \mathcal{S}} \log q\Big(x^{i_j} \mid x^{i_j-1}, x_t\Big) \\
&= \sum_{j=1}^{d} \log q\Big(x^{i_j} \mid x_t\Big) + \sum_{j\,:\,i_j - 1 \in \mathcal{S}} \Big[\log q\Big(x^{i_j} \mid x^{i_j-1}, x_t\Big) - \log q\Big(x^{i_j} \mid x_t\Big)\Big] \\
&= \sum_{i \in \mathcal{S}} \log q\Big(x^i \mid x_t\Big) + \sum_{i\,:\,i,\,i-1 \in \mathcal{S}} \mathrm{pmi}\Big(x^{i-1};\, x^i \mid x_t\Big),
\end{aligned} \tag{6}$$

where the second sum uses $\mathrm{pmi}(x^{i-1}; x^i \mid x_t) = \log q(x^i \mid x^{i-1}, x_t) - \log q(x^i \mid x_t)$. The first sum is over marginals that the denoiser is trained to match, so we may replace $q(x^i \mid x_t)$ by $p^i_{\theta,t}(x^i)$. Adding the cardinality bonus $c_0|\mathcal{S}|$ to both sides and grouping terms by position gives

$$c_0\,|\mathcal{S}| + \log q\Big(x^{\mathcal{S}} \mid x_t\Big) = \sum_{i \in \mathcal{S}} \underbrace{\Big[c_0 + \log p^i_{\theta,t}\Big(\hat{c}^i\Big)\Big]}_{= h_t^{i,\hat{c}^i}} + \sum_{i\,:\,i,\,i-1 \in \mathcal{S}} \underbrace{\mathrm{pmi}\Big(\hat{c}^{i-1};\, \hat{c}^i \mid x_t\Big)}_{= J^{i-1,i}}. \tag{7}$$

It remains to match the feasible sets. Identifying a set with its indicator vector, the map $\mathcal{S} \mapsto z$ with $z^i = 1$ iff $i \in \mathcal{S}$ is a bijection between subsets of $\mathcal{M}_t$ and points of $\{0,1\}^{|\mathcal{M}_t|}$, and under it the two sums in Equation 7 are exactly the linear and quadratic terms of Equation 4. The two problems therefore range over the same feasible set and agree in objective value at every point, hence share their maximizers. □

Remark (consistency with the already-committed context). The derivation does not yield a term for a predecessor that is already unmasked. Such a predecessor is part of x_t, and the denoiser conditions on it, so the dependence is already reflected in $p^i_{\theta,t}$. Line 4 of Alg. 1 nevertheless allows an optional $\gamma\,\mathrm{pmi}(x_t^{i-1}; \hat{c}^i)$ to be added; Equation 7 satisfies the proposal at $\gamma = 0$. The motivation for keeping γ as a hyperparameter comes from the fact that a position whose predecessor is already committed does not participate in the quadratic term, so its decision reduces to the sign of $h_t^{i,c}$ alone; the extra term lets the bigram prior influence that decision instead of leaving it entirely to the conditional marginal and the threshold c_0.

Remark (estimating the coupling). The derivation yields the pmi conditioned on the current state, $\mathrm{pmi}(\hat{c}^{i-1}; \hat{c}^i \mid x_t)$, which varies across steps and prompts. We use a static bigram model, dropping the conditioning on x_t for a corpus-level average. This is a table lookup, adding no forward pass or learned parameters, and Sec. 4 shows it already suffices to move the NFE–accuracy Pareto front. A context-aware estimator, which is e.g., read off the denoiser's hidden states, but signed and indexed by candidate values, would distinguish a pair that is improbable in general from one improbable here, and plugs into J unchanged. We leave this for future work.

Remark (multiple candidates per position). For $|\mathcal{C}^i| > 1$, the values x^i are no longer pinned to $\hat{c}^i$, so Equation 3 is maximized jointly over the set and its assignment, $(\mathcal{S}, v^{\mathcal{S}})$, and every x^i above is read as v^i. The final identification is then between feasible z and such pairs: the constraint $\sum_c z^{i,c} \leq 1$ makes $\mathcal{S} = \{i : \exists c,\ z^{i,c} = 1\}$ together with $v^i = c$ iff $z^{i,c} = 1$ a bijection, under which the bracketed unary becomes $h_t^{i,c}$ and the pairwise term $J^{i-1,i}(c', c)$, evaluated at $c = v^i$ and $c' = v^{i-1}$.

C Experimental details

C.1 Sudoku

Architecture and training. The denoiser is a 3-layer bidirectional Transformer (d=384, 12 heads, $\sim$5.4M parameters) over the 81 cells, with no causal mask and no explicit timestep embedding, since the number of masked cells implicitly encodes t. Following Mohamud et al. (2026), each answer cell of a clean solution is independently masked with probability $t \sim \mathcal{U}(0, 1)$ while clue cells remain untouched. At least one cell is masked per training example. We train with average cross-entropy loss over the masked positions under the standard MDM time weighting $1/t$, using AdamW (lr=10^{-3}, cosine decay, 500-step warmup, β=$(0.9, 0.95)$, weight decay 0.01), batch size 1024, for 50 000 steps.

We evaluate the same checkpoint on 100k puzzles in- and out-of-distribution, analogously to Kim et al. (2025a). **Easy** is the held-out split of Shah et al. (2024)'s filter over the dataset by Radcliffe (2020), covering puzzles that can be solved without backtracking. **Hard** is the hardest decile of the complement. All methods share the commit schedule, i.e., commit the same number of tokens per step, and see identical puzzles.

Greedy decoding baseline. Since no prior method combines consistency checking with confidence-ordered decoding, we construct a greedy baseline to isolate the contribution of solving Equation 1 to optimality. It builds the same candidate set as ConGO but resolves conflicts in a single sweep: positions are visited in decreasing confidence and each candidate consistent with those already committed is taken. It thus shares ConGO's value-awareness and differs only in that commitments are sequential and never revisited, so it cannot trade a high-confidence choice for one that unlocks many commitments elsewhere.

ConGO setup. The coupling J is a hard binary consistency check. Rather than penalizing violations, we impose them as constraints and solve Equation 1 as a mixed-integer linear program (MILP). While the same effect can be achieved by setting λ to a sufficiently high value, we find that encoding constraints explicitly yields better results in terms of end-to-end accuracy and solve time. Let y denote the known grid (clues and all committed tokens), $\mathcal{M}_t$ the masked cells, $\mathcal{U}$ the 27 units (9 rows, 9 columns, 9 boxes), and $P(i)$ the 20 peers of cell i (i.e., the cells that are in the same row, column or block as cell i). We introduce one binary variable $z^{i,c}$ per masked cell i and shortlisted value $c \in \mathcal{C}^i$, where $\mathcal{C}^i$ is the denoiser's top-k value at i with any value already present among $y_{P(i)}$ removed at build time. One program is built and solved per denoising step:

$$\max_{z} \sum_{i \in \mathcal{M}_t} \sum_{c \in \mathcal{C}^i} z^{i,c} \quad \text{s.t.} \quad \sum_{c \in \mathcal{C}^i} z^{i,c} \leq 1 \qquad \forall\, i \in \mathcal{M}_t, \tag{8}$$

$$\sum_{i \in u \cap \mathcal{M}_t} z^{i,c} \leq 1 \qquad \forall\, u \in \mathcal{U},\ c \in \{1, \ldots, 9\}, . \tag{9}$$

Row Equation 8 is the shortlist constraint of Equation 1 constraining each cell to either stay masked, or committed at exactly one value. It becomes active at $|\mathcal{C}^i| > 1$, where the solver may take a cell's second- or third-choice value if that admits further commitments elsewhere. Row Equation 9 encodes Sudoku's exclusion rules, one per (unit, value) pair, i.e., for cells in each row, column and square, there can be at most one cell filled with a specific value. As consistency with y is already enforced by the shortlist filter, only interactions among cells decided in the same step remain. Finally, for a given maximum number of steps T, each step commits at most $k_t = \lceil |\mathcal{M}_t|/(T - t) \rceil$ cells, the same linear schedule the baselines follow. We enforce this after the solve: the program returns a maximum consistent set, which is then ranked by $\log p^i_{\theta,t}(c)$ and truncated to k_t. The budget can equivalently be stated in the model as $\sum_{i,c} z^{i,c} \leq k_t$, in which case the solver returns the best consistent subset of that size rather than a maximum set truncated afterwards.

We use $h_t^{i,c} = 1$, so the objective counts commitments and the program is exactly the maximum consistent subset problem; the marginals then enter only through the shortlist selection. Programs are solved with Gurobi (Gurobi Optimization, LLC, 2026) via `gurobipy`. Since the absorbing process admits no un-commit move, a step whose optimum is empty would stall the sampler, so we commit the single most confident cell in that case (line 7 of Alg. 1).

Evaluation protocol. At test time, the model is queried iteratively: at each of T steps it scores every currently masked cell, commits a subset, and repeats until the grid is filled. Methods compared at a given T are matched on (i) the number of denoiser forward passes and (ii) the number of cells committed per step, so they differ only in which cells are committed. Batch size, checkpoint, and puzzle subsample are identical across methods. Wall-clock figures are end-to-end per puzzle. Timing measurement is performed for inference on CPU on Apple M4 Max, while training was done on an NVIDIA H100 GPU.

C.2 CODE GENERATION

With no external constraints available, the coupling is estimated from data. Selection is per semi-autoregressive block (of size 64) and positions are block-local, so the first position of each block carries no predecessor term. We take one candidate per position, $\mathcal{C}^i = \{\hat{c}^i\}$ with $\hat{c}^i = \arg\max_c p^i_{\theta,t}(c)$, collapsing the assignment variables to position indicators $z^i := z^{i,\hat{c}^i}$. All pmi values are read from a static bigram model built once from CodeParrot[1]. tokenized with the LLaDA tokenizer such that token IDs match the denoiser's output. We count unigram and adjacent-pair occurrences, resetting at document boundaries so no pair spans two documents. The pmi scores are then queried by lookup, adding no forward pass at inference.

ConGO-PMI setup. The objective terms are

$$h_t^i = c_0 + \log p^i_{\theta,t},$$

$$J^{i-1,i} = \lambda \left(\mathrm{pmi}\left(\hat{c}^{i-1};\, \hat{c}^i\right) - \theta_{ref} \right),$$

with $J^{i,j} = 0$ for all non-adjacent pairs. Whenever $\mathrm{pmi}\left(\hat{c}^{i-1};\, \hat{c}^i\right) < 0$, a hard constraint is added to prevent co-committing the corresponding tokens as they are considered inconsistent. Parameter θ_{ref} is an optional parameter to correct for the bigram model bias, and in all our experiments it was set to 0.1. We use $\lambda = 0.1$ for HumanEval and $\lambda = 0.3$ for MBPP. The value used for HumanEval was adjusted from the one obtained on the MBPP training set by scaling based on the median value of the unary coefficient on 3 HumanEval code generations, without checking their outcome.

ConGO-PMI-DAPD setup. The candidates are the masked positions of the current block, excluding the tokens that DAPD-direct commits unconditionally (confidence 1). The dependency of two positions is obtained by thresholding the dependency score $s^{i,j}$ of DAPD. The linear term is DAPD's confidence-weighted degree, normalized over the candidates,

$$h_t^i = \frac{d^i}{\max_k d^k} \cdot \frac{\mathrm{conf}_t^i}{\max_k \mathrm{conf}_t^k}, \qquad d^i = \textstyle\sum_j s^{i,j}, \quad \mathrm{conf}_t^i = \max_c p^i_{\theta,t}(c).$$

As the unary score is normalized, we also use the normalized pmi for coupling, $\mathrm{npmi}(c; c') = \mathrm{pmi}(c; c')/(-\log b(c, c')) \in [-1, 1]$, estimated from the bigram model, and set

$$J^{i,j}(\hat{c}^i, \hat{c}^j) = \begin{cases} \lambda\big(\mathrm{npmi}(\hat{c}^i; \hat{c}^j) - \theta_{\mathrm{ref}}\big) & \text{if } j = i+1,\ \mathrm{npmi}(\hat{c}^i; \hat{c}^j) \geq \theta_{\mathrm{veto}}, \\ -\infty & \text{if } (i,j) \text{ dependent or } j = i+1 \text{ and } \mathrm{npmi}(\hat{c}^i; \hat{c}^j) < \theta_{\mathrm{veto}}, \\ 0 & \text{otherwise.} \end{cases}$$

The offset $\theta_{\mathrm{ref}} = 0.1$ corrects for the positive bias of the bigram model, and $\theta_{\mathrm{veto}} = 0$ treats negatively associated pairs as inconsistent. We use $\lambda = 0.5$.

In both cases, the parameters were tuned on the training split of MBPP. An additional hard constraint is added to prevent committing more than 3 consecutive tokens. While it has been observed to not affect results significantly, it was used as a precaution to bound the length of co-committed chains as a result of relying on the first-order approximation. The resulting constrained binary quadratic programs were solved exactly with Gurobi (Gurobi Optimization, LLC, 2026).

Evaluation protocol. For HumanEval we report pass@1 under zero-shot evaluation, and for MBPP pass@1 under 3-shot evaluation with the harness's fixed prompt examples, consistent with the DAPD protocol. MBPP is evaluated on the 500-problem test split. The step budget of 256 corresponds to 64 steps in each of four semi-autoregressive blocks. All measurements are performed on a single NVIDIA H100 GPU.

[1] `https://huggingface.co/datasets/transformersbook/codeparrot`

Table 3: Exact solve rate (%) on Sudoku for varying NFEs and solve time (seconds) for 21 NFE on an Apple M4 Max. $|\mathcal{C}^i|$ is the number of candidate values per position.

	Easy (in-distribution)				Hard (out-of-distribution, top decile)			
	Accuracy (%) @NFE			Solve time (s)	Accuracy (%) @NFE			Solve time (s)
Method	5	9	21	@21 NFE	5	9	21	@21 NFE
Random order	0.5	1.4	2.7	**0.001**	0.1	0.4	1.0	**0.001**
Max probability	28.4	64.9	85.4	**0.001**	4.3	14.1	26.4	**0.001**
Probability margin	28.5	65.2	<u>85.5</u>	**0.001**	3.8	12.8	24.9	**0.001**
Negative entropy	23.9	62.1	84.5	**0.001**	3.2	13.2	<u>26.8</u>	**0.001**
Greedy	<u>29.8</u>	<u>65.4</u>	84.8	0.003	<u>4.8</u>	<u>14.6</u>	25.8	0.003
ConGO, $\lvert\mathcal{C}^i\rvert$=1	31.4	66.4	85.2	0.021	5.4	15.4	26.6	0.020
ConGO, $\lvert\mathcal{C}^i\rvert$=2	54.6	79.6	90.8	0.037	13.9	25.3	35.7	0.044
ConGO, $\lvert\mathcal{C}^i\rvert$=3	**81.4**	**93.3**	**97.6**	0.156	**45.3**	**59.8**	**69.2**	0.101

Table 4: One commit per step ($T = 60$) on 1000 held-out in-distribution Sudoku puzzles.

Method	Full Puzzle Accuracy (%)	Cell Accuracy (%)
Greedy	92.4	98.0
ConGO, $\lvert\mathcal{C}^i\rvert = 1$	92.6	98.0
ConGO, $\lvert\mathcal{C}^i\rvert = 3$	**99.2**	**99.7**

D Additional Results

D.1 Sudoku

D.1.1 MDM

In Table 3, we provide the extension of Table 2 with average sequence generation times. Although ConGO significantly reduces NFEs and improves accuracy, its runtime in the current configuration exceeds the cost of the saved model evaluations, as each forward pass of the considered 5.4M-parameter model is inexpensive. The introduced overhead of ConGO is amortized at larger model size, as shown by our UDM generalization in Table 5, as well as language generation experiments in Table 2.

Table 4 fixes the commit budget at one cell per step, so all methods run the same number of denoiser calls and commit in the same number of steps. Parallelism is thereby removed from the comparison, and what remains is the selection rule alone: which cell is committed, and at which value. At $|\mathcal{C}^i| = 1$ ConGO can only accept or reject each cell's arg max, so it reduces to picking the most confident consistent cell, and hence matches the greedy baseline. The gain at $|\mathcal{C}^i| = 3$ is therefore attributable entirely to value selection: given the option, the optimizer commits a cell at its second- or third-choice digit when the arg max is inconsistent with the current grid, rather than committing the arg max and propagating the error to a state from which no valid completion exists. This isolates the mechanism behind the parallel-decoding results of Sec. 4.1.

D.1.2 UDM

Training details. We finetune DiffusionGemma-26B-A4B-it (DiffusionGemma Team et al., 2026) with LoRA (Hu et al., 2022) using the Unsloth Sudoku recipe[2]. Training data is generated procedurally: we sample a completed grid by randomized backtracking, then remove cells one at a time, keeping a removal only if the puzzle still admits a unique solution, until 36–46 clues remain. The model is prompted with the grid as an 81-character string with 0 marking a blank, and trained to emit the completed 9×9 grid. Held-out puzzles are drawn from the same generator at disjoint seeds, so evaluation is in-distribution with respect to training but on grids the model has not seen.

ConGO setup. The formulation is the one of App. C.1, applied to a uniform-state model. Since UDMs re-noise at every step, a cell the program leaves unassigned is not frozen for the following steps. Therefore, no progress-guarantee fallback is needed and no commit budget is imposed, and every cell the solver selects is written. Second, consistency is checked against the static clue grid rather than a growing committed grid. We

[2] `https://unsloth.ai/docs/models/diffusiongemma`

Table 5: Held-out evaluation ($n = 200$) on finetuned DiffusionGemma-26B-A4B-it under maximum step budgets of 1, 16, and 64. Our ConGO is compared against the default and top-k baselines.

	Accuracy (%) ↑			Average NFE ↓			Solve time (s) ↓		
NFE budget	1	16	64	1	16	64	1	16	64
Top-k (default)	69.0	78.0	78.0	1.0	3.6	5.8	0.16	0.36	0.52
ConGO, $\lvert\mathcal{C}^i\rvert = 1$	69.0	78.0	78.5	1.0	3.3	**3.9**	**0.15**	**0.33**	**0.38**
ConGO, $\lvert\mathcal{C}^i\rvert = 3$	69.0	**83.0**	**82.5**	1.0	**3.2**	4.6	0.20	0.51	0.72

report two variants: $|\mathcal{C}^i| = 1$, where the program only filters the model's arg max per cell, and $|\mathcal{C}^i| = 3$, where it jointly selects cells and values from the three highest-probability candidates.

Evaluation. We evaluate on 200 held-out puzzles, keeping the denoiser frozen and varying only the acceptance rule. The baseline is the model's built-in entropy-bound sampler, i.e. the EB rule of Ben-Hamu et al. (2025). All strategies run under maximum budgets of 1, 16, and 64 denoising steps with the model's adaptive stopping left at its default, so the number of steps actually executed may fall below the budget. We therefore report the average steps run alongside the budget. A puzzle counts as solved only on an exact match against the full ground-truth grid, parsed from the first 81 digits of the decoded output. Wall-clock time is end-to-end per puzzle. Results are provided in Table 5.

D.2 Code Generation

Figure 3 extends Figure 2 with all evaluated configurations and the staged variant of DAPD. The two DAPD variants occupy different regimes. DAPD-staged requires 80–190 decoding steps, whereas the direct variants and ConGO operate within 20–60 steps, hence DAPD-staged was omitted in the main text. The additional value-blind commitments of the staged rule do not pay off on HumanEval, where its best accuracy (44%) matches that of DAPD-direct at more than twice the number of steps. On MBPP, it gains about 3 pp over the best direct configuration, but at roughly three times the number of steps. In the low-step regime, both ConGO variants shift the fronts of their respective baselines. ConGO-PMI dominates Fast-dLLM across the swept range, and ConGO-PMI-DAPD extends the DAPD-direct front towards fewer steps. The value-aware coupling therefore offers a cheaper route to co-committing consistent tokens than the staged rule's value-blind threshold.

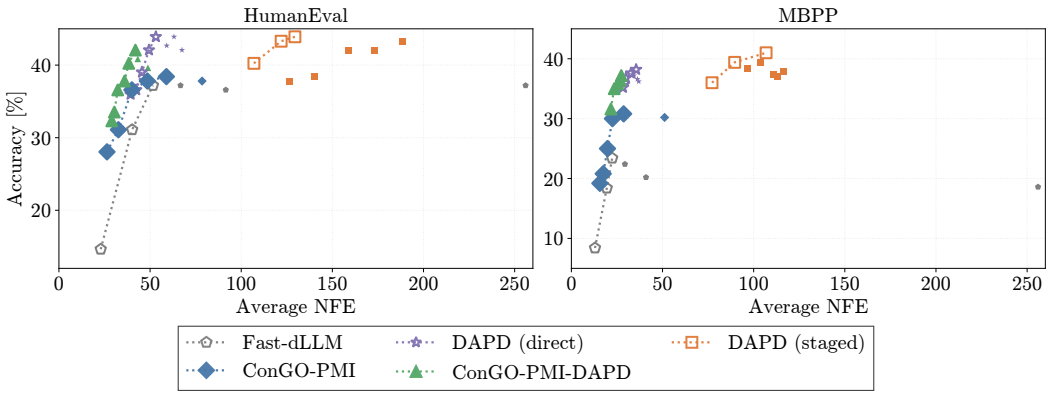

Figure 3: Extended version of Figure 2 with all evaluated configurations and the additional DAPD-staged baseline, on HumanEval and MBPP with LLaDA-8B-Instruct. Large markers connected by dotted lines form the Pareto front of each method; small markers are dominated configurations. For Fast-dLLM, the confidence threshold is swept over $\beta \in \{1, 0.9, 0.8, 0.7, 0.6, 0.5\}$, and for ConGO-PMI the corresponding $c_0 = -\log \beta$. For DAPD-direct, DAPD-staged, and ConGO-PMI-DAPD, the dependency threshold is swept over $\tau_{\max} \in \{0.01, 0.02, 0.035, 0.05, 0.075, 0.10, 0.15, 0.20, 0.30\}$.