MULTI-HOP ATTENTION GRAPH SAMPLING FOR MASKED DIFFUSION LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Masked Diffusion LLMs can accelerate language generation through parallel decoding, but unmasking incompatible token combinations can degrade quality. Recent attention-based dependency-aware samplers construct a graph where the masked positions are nodes and the edges correspond to direct attention map weights. Though achieving impressive acceleration, these consider only direct or single-hop relations in the graph, disregarding multi-hop connectivity. We introduce GRID – *Graph Resistance Independent Decoding* – a training-free sampler that replaces the direct-edge coupling score with *effective resistance*, a parameter-free connectivity measure that aggregates both direct and indirect routes between masked positions. To analyze how samplers perform on complex multi-hop graph structures, we introduce *Dep-Bench*, a controlled benchmark with five datasets spanning four graph structures and known dependencies at varying depths. On Dep-Bench, GRID's advantage over direct-edge methods grows consistently with depth. On ParallelBench, a public benchmark focused on parallel decoding, GRID achieves up to $\times 2$ decoding-step speedup over the fastest competing sampler, for a given quality threshold of 95% of the category-level top-1 score. Finally, across 16 leading model and LLM reasoning benchmark combinations, GRID achieves $\times 3$–10 speedup over top-1 decoding and 7–11% improvement over the fastest competing sampler, at comparable accuracy. The combined results validate that multi-hop connectivity is a valuable and underutilized signal for dependency-aware parallel decoding.

1 INTRODUCTION

Masked diffusion models (MDMs) (Austin et al., 2021a; Lou et al., 2023; Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2025) such as LLaDA (Nie et al., 2026; Bie et al., 2025) and Dream (Ye et al., 2025) retain language modeling quality competitive with auto-regressive (AR) models, while offering high speedup potential through parallel decoding. In contrast to AR models that predict the next-token distribution, MDMs provide a conditional marginal distribution for each masked token at a given denoising step. Decoding multiple masked tokens in parallel implicitly approximates their joint conditional distribution by a factorized product of these per-token marginals. When the selected tokens are coupled, this approximation can lead to *joint-marginal mismatch* (Kim et al., 2026a; Luo et al., 2026): each token may be plausible given the partially masked context, but the unmasked combination is misaligned. For instance, the response of $[m_1, m_2] = [\textit{Rome}, \textit{France}]$ to "$[m_1]$ *is the capital of* $[m_2]$*"* is independently plausible but jointly incorrect. Identifying independent tokens to unmask in parallel thus becomes the main lever for speedup in diffusion-based samplers.

Training-free diffusion samplers mostly utilize individual token signals (Wu et al., 2025; Ben-Hamu et al., 2026; Kim et al., 2026b; Zhang et al., 2026; Luxembourg et al., 2025), such as confidence and entropy, to determine which tokens to simultaneously unmask. While providing increased speedup, these methods fail to model the pairwise relations between masked tokens, leading to degraded quality due to the joint-marginal mismatch. Subsequent dependency-aware samplers (Azangulov et al., 2026; Ringel et al., 2026) estimate the pairwise coupling between masked tokens using different proxies, one of them being the model's self-attention. Attention-derived methods, such as DAWN (Luo et al., 2026) and DAPD (Kim et al., 2026a), construct a graph where the nodes and edges correspond to masked tokens and their pairwise attention map values respectively. Weak direct edges suggest weaker coupling between the corresponding masked tokens, making those can-

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.

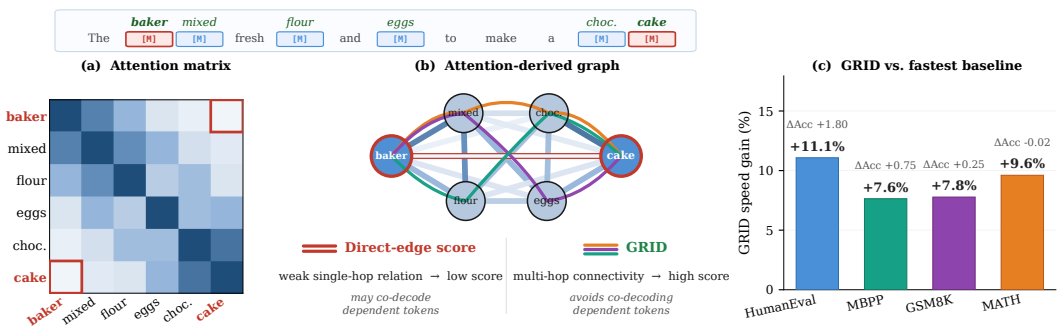


Figure 1: **GRID – intuition and results highlights.** In the sentence above: the *baker* and *cake* nodes have a low direct-attention coupling score, as shown in *(a)*, but looking at the dependency graph reveals additional strong multi-hop relations through intermediate edges. Our proposed sampler, GRID (**G**raph **R**esistance **I**ndependent **D**ecoding), offers a richer connectivity measure, considering both direct/single-hop and multi-hop relations in the graph. In *(c)*, our training-free dependency-aware sampler, evaluated over 16 combinations of models and benchmarks, provides a considerable decoding speedup versus the fastest competitor at comparable accuracy.

didates for parallel decoding. Although improving quality over individual-token techniques, such dependency-aware samplers consider mostly direct single-hop relations in the graph, while disregarding complex multi-hop routes as shown in Figure 1 *(a)–(b)*.

We suggest that multi-hop paths in the attention-derived graph carry relevant information about complex dependencies that direct edges lack (see Figure 1*(b)*), motivating a better proxy that captures broader graph connectivity. Effective resistance (ER), a parameter-free classical measure in graph theory (Klein & Randić, 1993), models the graph as an electrical circuit where nodes that have multiple paths between them benefit from better electrical flow and lower effective resistance, and vice versa. We thus introduce *GRID*—**G**raph **R**esistance **I**ndependent **D**ecoding—a training-free sampler that utilizes the normalized inverse ER on the attention-derived graph as a generalized multi-hop pairwise coupling score. Building on DAWN's confidence-guided selection procedure (Luo et al., 2026), GRID uses its ER-based coupling score rather than direct attention edges to mitigate conflicts, then greedily selects a confidence-ordered independent set of masked tokens for parallel decoding. The complete procedure is given in Algorithm 1.

We first motivate inverse ER as a multi-hop connectivity proxy through its exact expansion into direct and multi-hop connectivity contributions (Section 3). We then introduce *Dep-Bench*, a controlled benchmark with five datasets and four task-dependency structures whose dependencies are known by construction, enabling a controlled analysis as a function of dependency hop distance and structure (Section 5.1). On Dep-Bench, GRID's advantage over direct-edge attention increases with dependency hop distance. We next evaluate on ParallelBench (Kang et al., 2026), a public benchmark for speed–quality trade-offs in parallel decoding. At a common quality threshold of 95% of the category-level top-1 score (one-position-per-step decoding), GRID achieves up to $\times 2$ decoding-step speedup over the fastest competing sampler across the three categories (Section 5.2). Finally, across four MDMs and four standard code and math benchmarks, GRID achieves the highest decoding speed for all 16 model–task settings, improving over the fastest competing sampler by 7.6%–11.1% on average while maintaining comparable accuracy (Section 5.2).

2 Related Work

MDM parallelization can be tackled using two approaches, one that targets the model and trains it to provide fewer decoding steps, and the second that optimizes over inference-time sampling. The works of Di4C (Hayakawa et al., 2025), SDTT (Deschenaux & Gulcehre, 2025), Duo (Sahoo et al., 2025), FS-DFM (Karimi Monsefi et al., 2025), dParallel (Chen et al., 2025b), and dUltra (Chen et al., 2025a) belong to the first category, training or distilling models for improved parallel decoding. Differently, training-free samplers use signals such as confidence, entropy, prediction stability, lookahead, or positional separation (Wu et al., 2025; Ben-Hamu et al., 2026; Kim et al., 2026b;

Zhang et al., 2026; Luxembourg et al., 2025), without explicitly estimating pairwise compatibility between positions decoded together.

Dependency-aware methods estimate such coupling directly. PUNT (Azangulov et al., 2026) probes the model with additional forward passes, while DEMASK (Ringel et al., 2026) trains a dependency predictor. Among attention-based methods, DAWN (Luo et al., 2026) and DAPD (Kim et al., 2026a) are particularly related: both use direct attention-derived coupling between masked positions, with a key difference being their selection procedure. CLAD (Qi et al., 2026) also derives compatibility signals from self-attention. These methods, however, do not explicitly aggregate connectivity through indirect paths in the attention-derived graph. GRID instead uses effective resistance to aggregate both direct and indirect connectivity. Appendix K provides additional discussion of related acceleration methods.

3 Pairwise Dependency and Multi-Hop Connectivity

Parallel decoding and pairwise dependency. Consider a MDM parameterized by θ at decoding step t, with a partially decoded sequence x_t of length N. Let $M \subseteq \{1, \dots, N\}$ denote the set of currently masked positions and x^i the unknown token at position i. For each $i \in M$, the model provides the conditional marginal

$$\mathbf{p}_i = p_\theta(x^i \mid x_t).$$

In parallel decoding, a set of positions $U \subseteq M$ is jointly committed using these per-position predictions, implicitly approximating their joint conditional distribution by a product of marginals,

$$p_\theta\big(\{x^i\}_{i\in U} \mid x_t\big) \;\approx\; \prod_{i\in U} p_\theta(x^i \mid x_t).$$

This factorization is exact when the positions in U are conditionally independent given x_t. When they are coupled, however, committing them simultaneously ignores information that one position could provide about another, potentially yielding mutually inconsistent predictions (Kim et al., 2026a; Luo et al., 2026).

For a pair of masked positions $i, j \in M$, this dependence can be probed by asking how the prediction at i changes after revealing j. Following prior work (Ringel et al., 2026; Azangulov et al., 2026), let

$$\hat{x}^{\,j} = \arg\max_v p_\theta(x^j{=}v \mid x_t),$$

where $\hat{x}^{\,j}$ is the highest probability token at position j given decoding state x_t, and let x_t^{+j} denote the state obtained by replacing the mask at j with $\hat{x}^{\,j}$.

Definition 1 (Counterfactual pairwise dependency)**.** *For $i, j \in M$, define*

$$K_{i\leftarrow j} \;\triangleq\; \mathrm{KL}\Big(p_\theta(x^i \mid x_t) \;\Big\|\; p_\theta(x^i \mid x_t^{+j})\Big),$$

where KL denotes the Kullback–Leibler divergence between $p_\theta(x^i)$, conditioned on given state x_t, before and after unmasking token x^j. A small $K_{i\leftarrow j}$ value means revealing j has little effect on the prediction at i, while a large value indicates stronger dependence. Computing the full counterfactual dependency matrix is expensive, as it requires $|M|$ additional forward passes per decoding step, one for each $j \in M$. We therefore seek a dependency proxy that requires no additional forward passes.

Effective resistance as a pairwise coupling score. Effective resistance is a well-known connectivity measure in graph theory, inspired by electrical circuits (Klein & Randić, 1993). Nodes that have multiple connecting paths through both single-hop and multi-hop connections promote improved electrical flow, and thus lead to low effective resistance. In the context of MDMs and attention-based sampling (Kim et al., 2026a; Luo et al., 2026), a directed graph is obtained from the attention weights at each forward pass. We additionally symmetrize the directed attention weights by averaging the two directions, yielding a connected undirected weighted graph with symmetric, nonnegative adjacency matrix $\mathbf{A}$. Here, A_{ij} denotes the weight of edge $\{i, j\}$, and $A_{ii} = 0$.

Let R_{ij} denote the effective resistance between nodes i and j. Denote the full vertex set by V, and let $S = V \setminus \{i, j\}$ denote the set after excluding nodes i and j. Let $d_k = \sum_{v\in V} A_{kv}$ be the weighted degree of node k. The inverse ER can be expanded as (more detail in Appendix C).

$$R_{ij}^{-1} = \underbrace{A_{ij}}_{\text{1-hop}} + \underbrace{\sum_{k \in S} \frac{A_{ik}A_{kj}}{d_k}}_{\text{2-hop}} + \underbrace{\sum_{k,\ell \in S} \frac{A_{ik}A_{k\ell}A_{\ell j}}{d_k d_\ell}}_{\text{3-hop}} + \underbrace{\cdots}_{\text{4+ hops}} . \tag{1}$$

As can be seen, the first term captures the single-hop/direct connection between i and j and the subsequent terms capture the two-hop, three-hop and N-hop connectivity through intermediate nodes. Thereby, even if two nodes have a weak direct edge, as portrayed in the *baker–cake* pair in Figure 1, inverse ER will still get a higher score contributed by multi-hop connectivity. Inverse ER can thus be a pairwise coupling score, where a high R_{ij}^{-1}, or equivalently low R_{ij}, corresponds to stronger connectivity between i and j. A full derivation of the above expansion is included in Appendix C.

Note there exist other graph measures that incorporate direct and indirect route connections, such as Katz Similarity (Katz, 1953) and Heat Kernel (Kondor & Lafferty, 2002). An empirical comparison in Appendix I.2 between the three reveals they can all serve as effective pairwise dependency scores, but ER is more appealing as it can be used out-of-the-box without tuning extra hyper-parameters, whereas Katz and Heat Kernel require tuning an additional decay or time-scale parameter.

While the previously defined counterfactual dependency $K_{i \leftarrow j}$ is directional, effective resistance on the undirected attention-derived graph is symmetric. Nonetheless, this symmetry is better suited for pairwise co-decoding, which requires a single compatibility score for each pair of positions. Inverse ER captures multi-hop connectivity in the attention-derived graph and is used here as a proxy for the model's underlying pairwise dependency between positions. We validate this proxy against counterfactual dependency as a model-grounded reference.

Counterfactual-dependency validation. To isolate the effect of the pairwise coupling score, we perform the following. At a given decoding state, we compute either direct-attention or inverse-ER scores over the same candidate masked positions, and use the same greedy rule to select a set of four positions. We then evaluate the dependency within each selected set using the counterfactual dependency from Definition 1. For a selected set $\mathcal{I}$, let $\mathcal{D}(\mathcal{I})$ denote the average over all pairwise symmetric dependencies $K_{ij}^{\text{sym}} = \frac{1}{2}(K_{i \leftarrow j} + K_{j \leftarrow i})$ among positions in $\mathcal{I}$. To measure how close the selected set is to the lowest-dependency candidate set, we define the *normalized dependency regret*

$$\text{Reg}(\mathcal{I}) = \frac{\mathcal{D}(\mathcal{I}) - \mathcal{D}_\star}{\mathcal{D}_{\text{avg}} - \mathcal{D}_\star},$$

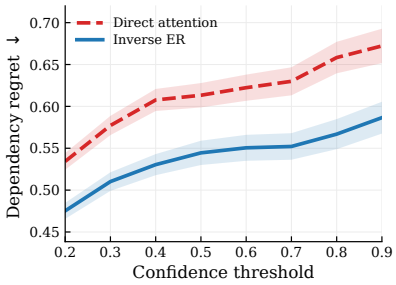


Figure 2: HumanEval with LLaDA-8B-Instruct. Shaded bands denote ±SEM across diagnostic states.

where $\mathcal{D}_\star$ is the minimum over all candidate four-position sets and $\mathcal{D}_{\text{avg}}$ is their mean. A normalized regret of zero is obtained for the optimal set, while one corresponds to the mean of a uniformly random candidate set. Figure 2 reports this regret on the HumanEval dataset (Chen et al., 2021) across different confidence thresholds. As can be seen, inverse-ER achieves lower regret at every evaluated threshold, and the same trend holds across the four standard code and math benchmarks evaluated in Section 5.2. The full protocol and remaining results are provided in Appendix B.

4 GRID: Effective-Resistance Parallel Decoding

We instantiate the inverse-ER proxy of Section 3 in a training-free parallel sampler, *GRID* (**G**raph **R**esistance **I**ndependent **D**ecoding). GRID builds on DAWN's attention aggregation, sink filtering, and confidence-ordered greedy selection procedure (Luo et al., 2026), replacing its direct-attention conflict rule with normalized inverse-ER coupling and a step-dependent coupling threshold. GRID thereby captures multi-hop relationships between masked positions. Algorithm 1 summarizes one denoising step.

Graph construction. Following DAWN, we average attention across heads of the final four transformer layers and filter attention sinks. The sink filter suppresses positions that attract broadly high

Algorithm 1 GRID: one denoising step

Input: masked positions M, marginals $\{\mathbf{p}_i\}_{i\in M}$, attention maps
Parameters: block size B, thresholds $\tau_{\text{high}}, \tau_{\text{low}}, q_0, \delta$
$\mathbf{A} \leftarrow$ head- and layer-averaged, sink-filtered, symmetrized attention restricted to M
Form $\widehat{\mathbf{A}}$ by removing the degree-expected background from $\mathbf{A}$ via Equation (2)
Compute effective resistance R_{ij} on $\widehat{\mathbf{A}}$ via Equation (3)
Compute pairwise couplings q_{ij} and coupling threshold $q_t^\star$ via Equation (4)
Select high-confidence positions $U \leftarrow \{i \in M : \text{conf}(i) \geq \tau_{\text{high}}\}$
Form candidate set $C \leftarrow \{i \in M \setminus U : \text{conf}(i) \geq \tau_{\text{low}}\}$
$S \leftarrow$ greedily select non-conflicting positions from C in decreasing confidence order
    subject to $q_{ij} \leq q_t^\star$ for all distinct $i, j \in S$
$U \leftarrow U \cup S$
if $U = \varnothing$ **then**
    $U \leftarrow \{\arg\max_{i\in M} \text{conf}(i)\}$
end if
return U

attention and can otherwise dominate pairwise scores. We then zero the diagonal and restrict the graph to the currently masked positions M. Let $\mathbf{A} \in \mathbb{R}^{|M|\times|M|}$ denote the resulting masked-position attention matrix. We symmetrize it by averaging the two directions, $\mathbf{A} \leftarrow (\mathbf{A} + \mathbf{A}^\top)/2$, yielding a symmetric weighted adjacency matrix. Because attention is computed over the full sequence before this restriction, prompt tokens and already-unmasked positions still affect the attention weights, but are not retained as vertices in the resistance computation.

Effective-resistance computation. Before computing ER, we apply a modularity-based correction to remove the degree-expected background from the attention graph. For $d_i = \sum_{j\in M} A_{ij}$ and $\text{vol}(\mathbf{A}) = \sum_{i\in M} d_i$, define

$$\widehat{A}_{ij} = \max\left(A_{ij} - \frac{d_i d_j}{\text{vol}(\mathbf{A})}, 0\right). \tag{2}$$

Here $d_i d_j/\text{vol}(\mathbf{A})$ is the expected edge weight under the weighted configuration-model null used in modularity (Newman & Girvan, 2004; Newman, 2004). The correction retains only positive connectivity above this degree-expected background, without introducing an additional tunable edge threshold. We use this correction to prevent broadly connected, high-degree nodes from contributing through many indirect paths and inflating the resulting pairwise coupling under ER. We then compute ER on the denoised graph $\widehat{\mathbf{A}}$. Rather than evaluating the expansion in Equation (1) directly, we use the equivalent Laplacian representation. Let $\mathbf{D}$ denote the degree matrix of $\widehat{\mathbf{A}}$, with $D_{ii} = \sum_j \widehat{A}_{ij}$, and define the graph Laplacian $\mathcal{L} = \mathbf{D} - \widehat{\mathbf{A}}$. ER is then computed as

$$R_{ij} = \mathcal{L}^\dagger_{ii} + \mathcal{L}^\dagger_{jj} - 2\mathcal{L}^\dagger_{ij}, \tag{3}$$

where $\mathcal{L}^\dagger$ is the Moore–Penrose pseudoinverse.

Effective-resistance coupling. GRID uses inverse-ER as a pairwise coupling score, where two candidates are deemed coupled if they surpass a given threshold. As positions are committed, the graph changes in size and overall edge-weight scale, so we instead compute the normalized coupling measure using the mean weighted degree $\bar{d} = \text{vol}(\widehat{\mathbf{A}})/|M|$ and the step-dependent threshold $q_t^\star$

$$q_{ij} = \frac{R_{ij}^{-1}}{\bar{d}}, \qquad q_t^\star = q_0 + \delta\left(1 - \frac{|M|}{B}\right), \tag{4}$$

where B is the block size, $|M|$ the current number of masked tokens, q_0 the initial coupling threshold, and δ controls how much the threshold increases as the block empties. Larger q_{ij} indicates stronger coupling, and two candidate positions are treated as conflicting when $q_{ij} > q_t^\star$. The $\bar{d}$ normalization removes uniform graph-scale drift across decoding steps. Appendix G provides the corresponding scale-invariance argument. As $|M|$ decreases, $q_t^\star$ increases from q_0 toward $q_0 + \delta$, gradually relaxing the coupling criterion and allowing greater parallelism as more context becomes available.

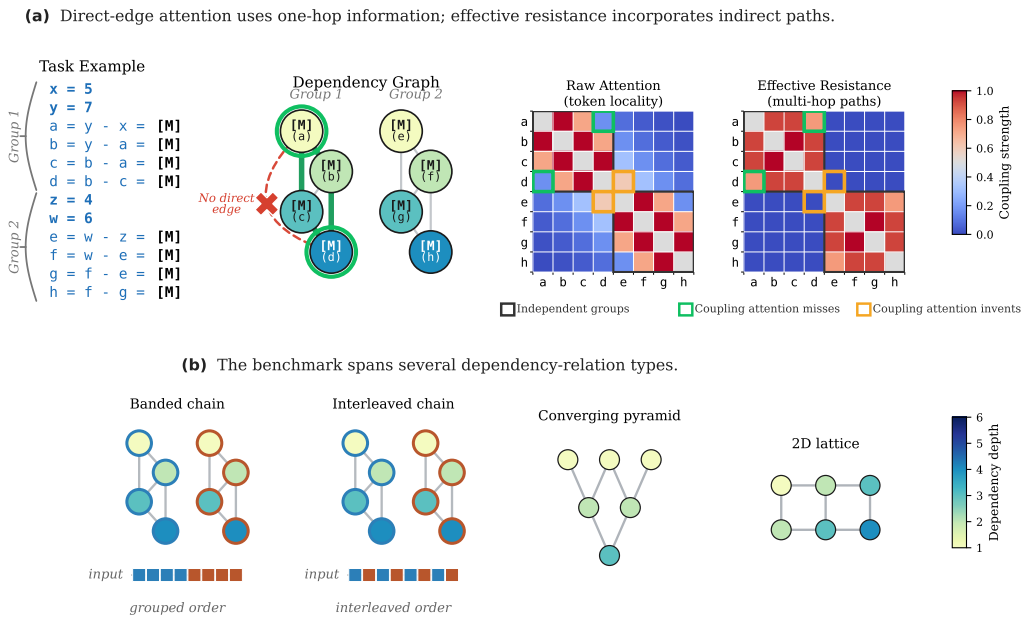


Figure 3: **Overview of Dep-Bench.** *(a)* Illustrative arithmetic-chain example, its ground-truth dependency graph, and direct-attention versus inverse-ER coupling maps. The highlighted pair has no direct edge but is connected by multiple indirect paths. Appendix A shows the corresponding coupling maps computed from LLaDA-8B-Instruct's actual attention maps on a real Dep-Bench example. *(b)* The four dependency structures used in the benchmark; node color denotes dependency depth.

Position selection. GRID follows DAWN's confidence-gated greedy selection procedure. Let $\mathrm{conf}(i) = \max_v [\mathbf{p}_i]_v$ denote the confidence of masked position i. Positions with $\mathrm{conf}(i) \geq \tau_{\mathrm{high}}$ are committed directly. From the remaining positions with $\mathrm{conf}(i) \geq \tau_{\mathrm{low}}$, GRID considers candidates in descending confidence order and greedily selects each candidate that does not conflict with any candidate already selected in this stage. If neither stage selects a position, GRID commits the most-confident masked position. Appendix D provides the full implementation details.

5 Results

We first introduce *Dep-Bench*, a controlled benchmark whose task-dependency structure is known by construction, allowing us to study how decoding performance changes with dependency hop distance (Section 5.1). We then evaluate GRID on ParallelBench (Kang et al., 2026) and on standard code and math benchmarks across multiple MDMs (Section 5.2).

5.1 Dep-Bench: Controlled Dependency Benchmark

Standard end-to-end (E2E) benchmarks do not expose ground-truth dependency structures at controlled dependency depths, making it difficult to isolate how samplers handle multi-hop dependencies. We therefore introduce *Dep-Bench*, a masked-infill benchmark with dependencies known by construction across four structures and controlled dependency depths (Figure 3). We use dependency depth to describe how far a masked position lies downstream in the known dependency graph. For example, in Figure 3*(a)*, a has depth 1 because it depends only on the provided values $x = 5$ and $y = 7$, while d has depth 4. This enables direct analysis of whether multi-hop scoring improves over direct attention as dependency depth increases.

The five datasets vary in dependency structure, token layout, and surface form, spanning banded and interleaved chains, a converging pyramid, and a 2D lattice (Figure 3). Each dataset contains 1500 examples with freshly sampled values and identifiers. Each example contains three independent

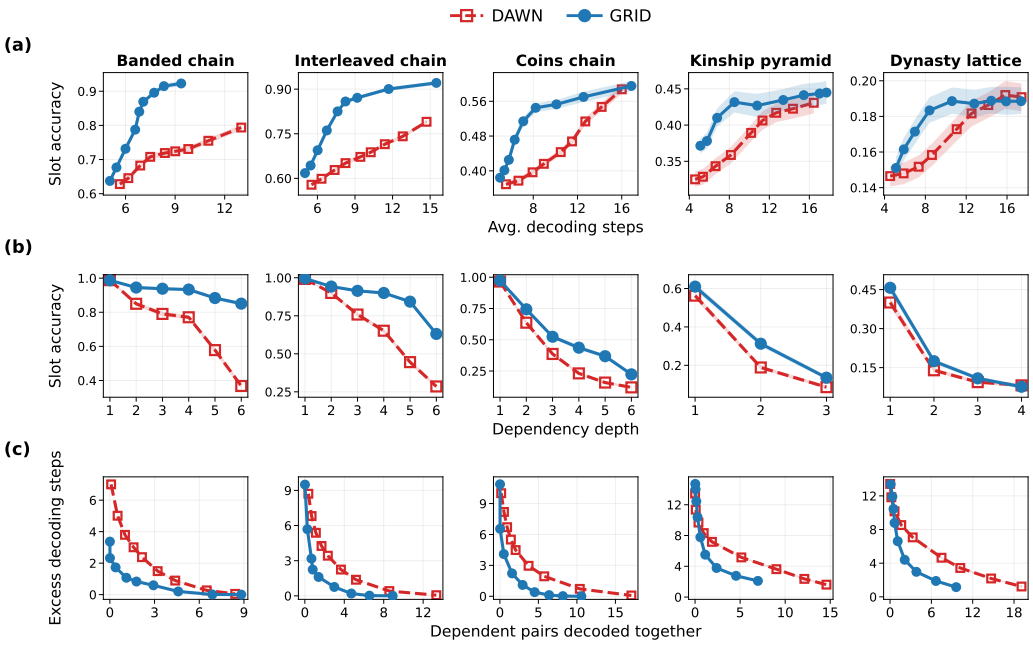


Figure 4: **Controlled DAWN–GRID comparison on Dep-Bench.** Results using LLaDA-8B-Instruct, with one dataset per column. *(a)* Slot accuracy vs. average decoding steps across dependency-threshold sweeps. *(b)* Slot accuracy versus dependency depth at a matched ≈9-step budget. *(c)* Trade-off between dependent co-decoding and excess serialization across dependency-threshold sweeps. Shaded bands are 95% bootstrap confidence intervals over examples.

groups of six masked single-digit answers (18 in total), with dependencies given by the dataset's known directed acyclic graph. Appendix A gives the exact constructions and worked examples.

To isolate the dependency-scoring mechanism, we compare simplified variants of GRID and DAWN (Luo et al., 2026) using the same confidence-ordered greedy selection procedure but different dependency scores: direct attention-edge coupling for DAWN and inverse ER for GRID. For each variant, we sweep its dependency-score threshold, which controls the level of parallelism. We use slot accuracy, the fraction of the 18 masked answers predicted correctly, to measure prediction quality. The controlled-comparison protocol is detailed in Appendix H.

The controlled comparison shows that GRID improves the accuracy–steps trade-off over DAWN, with a larger advantage at greater dependency depths, while also incorrectly co-decoding fewer dependent pairs and requiring fewer excess decoding steps. Figure 4*(a)* shows that GRID Pareto-dominates DAWN on all five datasets, achieving higher slot accuracy across the evaluated decoding-step budgets. At a matched ≈9-step budget, Figure 4*(b)* shows that the two methods are comparable at dependency depth $d = 1$, but the gap increases with depth. On Banded Chain, for example, accuracy at $d = 6$ is 0.85 for GRID versus 0.37 for DAWN, consistent with the multi-hop motivation of Section 3. Figure 4*(c)* evaluates decoding relative to the known Dep-Bench dependency graph. The x-axis measures the mean number of dependent pairs incorrectly co-decoded in the same step, and the y-axis measures the mean number of decoding steps beyond the graph-implied minimum. The origin therefore represents the optimal point, with zero dependent pairs incorrectly co-decoded and zero excess decoding steps. GRID's curves lie closer to the origin across all five datasets, indicating fewer dependent pairs incorrectly co-decoded and fewer excess decoding steps.

5.2 Decoding Performance on Public Benchmarks

We evaluate GRID on the recently introduced ParallelBench (Kang et al., 2026) dataset, as well as on four standard code and math benchmarks: HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021b), GSM8K (Cobbe et al., 2021), and MATH (Hendrycks et al., 2021). We use the following

MDMs throughout our evaluations: LLaDA-8B-Instruct (Nie et al., 2026), LLaDA 1.5 (Zhu et al., 2026), LLaDA 2 Mini (16B) (Bie et al., 2025), and Dream-v0-Instruct-7B (Ye et al., 2025).

We compare against top-1 decoding (one position per step), EB-Sampler (Ben-Hamu et al., 2026), Fast-dLLM (Wu et al., 2025), KLASS (Kim et al., 2026b), DAWN (Luo et al., 2026), CLAD (Qi et al., 2026) and PUNT (Azangulov et al., 2026). We include PUNT only on the evaluations of the standard LLM benchmarks as they report TPS and speedup, accounting for the additional forward passes that PUNT requires, whereas ParallelBench reports only decoding steps. DAWN (Luo et al., 2026) and DAPD (Kim et al., 2026a) both use direct attention-derived coupling between masked positions, differing primarily in their position-selection procedures. We use DAWN as the main direct-attention baseline, while DAPD is reported separately in Appendix F because its official implementation did not yield an accuracy-comparable operating point under our evaluation.

ParallelBench. ParallelBench (Kang et al., 2026) contains 17 tasks grouped into three categories: Waiting Line, Text Writing, and Puzzles. We merge the tasks within each category for category-level evaluation. For each sampler, we evaluate the parameter settings that control its speed–quality trade-off, applying each setting consistently across the entire category. At each quality target, we select the lowest-step setting that reaches the specified fraction of the category-level top-1 score (one-position-per-step decoding). We then compare GRID with the fastest competing sampler for each category and target. At the 95% target, GRID achieves matched-quality decoding-step speedups of $\times 1.06$ on Waiting Line, $\times 2.02$ on Text Writing, and $\times 1.31$ on Puzzles (Table 1). Additional protocol details, representative score–steps trade-offs, and robustness results are provided in Appendix E.

Table 1: **ParallelBench matched-quality step speedup** ($\times$, $\uparrow$) over the fastest competing sampler.

Category	90%	95%	99%
Waiting Line	$\times 1.02$	$\times 1.06$	$\times 0.98$
Text Writing	$\times 1.57$	$\times 2.02$	$\times 1.95$
Puzzles	$\times 1.65$	$\times 1.31$	$\times 1.36$

Standard code and math benchmarks. We follow DAWN's practice of tuning the hyper-parameters once per model on HumanEval, and applying the same configuration on the other three datasets. A one-at-a-time hyperparameter sensitivity analysis is provided in Appendix I.3, and full implementation details are given in Appendix D. Results in Table 2 show that GRID attains the highest speedup in all 16 model–benchmark settings. Averaged across models, its speed improvement over the fastest competing sampler ranges from 7.6% to 11.1%, with mean accuracy changes of $+1.80$, $+0.75$, $+0.25$, and -0.02 on HumanEval, MBPP, GSM8K, and MATH, respectively (see Figure 1*(c)*). GRID also uses the fewest decoding steps among single forward-pass samplers in all 16 settings. PUNT requires fewer nominal decoding steps in some settings, but uses additional model evaluations within each step, reducing throughput and speedup. DAWN is our closest structural baseline as we perform roughly the same confidence-guided position selection strategy, but replace the direct-attention coupling score with inverse-ER. Nevertheless, GRID achieves comparable accuracy with fewer decoding steps, across all 16 model-benchmark combinations.

Speed–quality analyses and ablations. To complement the single operating-point E2E results in Table 2, we explore multiple speed-quality tradeoffs. On ParallelBench, we sweep all competing parallel samplers and compare them at category-level matched-quality targets. Using the same simplified GRID–DAWN comparison as in Section 5.1, Appendix H repeats this analysis on the same four standard benchmarks as Table 2, where GRID achieves a consistently stronger accuracy–steps frontier, with the advantage increasing at tighter decoding budgets. Appendix I isolates the added effect of modularity denoising alone, versus the entire inverse-ER coupling score. Finally, we evaluate the performance of alternative multi-hop graph connectivity measures in Appendix I.2, specifically Katz similarity (Katz, 1953) and heat kernel (Kondor & Lafferty, 2002), and show they also improve over direct-edge attention, but with proper hyper-parameter tuning.

6 LIMITATIONS

A common limitation shared with other attention-based samplers is that direct attention is only a proxy for true masked token dependency. Although the counterfactual-dependency validation in Figure 2 supports inverse ER as a useful dependency proxy across the evaluated benchmarks, this

Table 2: End-to-end parallel decoding on four benchmarks across four MDMs. We report task accuracy (Acc, ↑), decoding steps (Steps, ↓), tokens per second (TPS, ↑), and speedup over top-1 decoding (Speedup, ↑). Light-gray rows denote dependency-aware samplers (DAWN, CLAD, PUNT, and GRID); PUNT uses additional model evaluations per decoding step. **Bold** marks the highest speedup in each model–benchmark setting, while underline marks the second highest.

	HumanEval				MBPP				GSM8K				MATH			
Method	Acc↑	Steps↓	TPS↑	Speedup↑	Acc↑	Steps↓	TPS↑	Speedup↑	Acc↑	Steps↓	TPS↑	Speedup↑	Acc↑	Steps↓	TPS↑	Speedup↑
LLaDA-8B-Instruct																
Top-1	42.1	256	5.90	×1.00	42.0	256	2.07	×1.00	78.7	256	2.06	×1.00	32.9	256	2.85	×1.00
EB-Sampler	37.2	59.5	25.39	×4.30	41.0	52.6	10.11	×4.88	78.2	57.9	9.15	×4.44	32.6	73.2	10.00	×3.51
Fast-dLLM	37.8	58.7	26.04	×4.41	42.0	51.5	10.23	×4.94	78.5	55.2	9.59	×4.66	32.0	71.8	10.19	×3.58
KLASS	39.6	104.9	14.5	×2.46	41.0	91.1	5.8	×2.80	74.6	101.0	5.2	×2.52	31.6	121.0	6.1	×2.14
DAWN	36.6	58.8	25.52	×4.33	41.3	53.7	9.93	×4.80	77.6	55.5	9.29	×4.51	32.3	72.6	9.93	×3.48
CLAD	37.8	60.0	25.27	×4.28	41.6	52.7	9.92	×4.79	79.3	55.5	9.30	×4.51	31.7	72.8	9.91	×3.48
PUNT	39.6	53.6	5.55	×0.94	39.8	44.1	2.11	×1.02	75.9	54.9	1.94	×0.94	33.3	64.5	2.29	×0.80
GRID (ours)	37.8	49.9	29.11	×**4.93**	42.4	43.4	11.34	×**5.48**	78.1	47.9	10.67	×**5.18**	32.1	62.3	11.46	×**4.02**
LLaDA-1.5																
Top-1	43.9	256	6.03	×1.00	42.6	256	2.25	×1.00	82.0	256	2.08	×1.00	34.8	256	2.87	×1.00
EB-Sampler	37.2	58.3	26.49	×4.39	40.2	56.4	10.38	×4.61	80.7	57.4	9.30	×4.47	33.4	70.9	10.36	×3.61
Fast-dLLM	36.6	55.3	27.75	×4.60	42.0	55.2	10.55	×4.69	81.2	54.9	9.70	×4.66	33.1	69.5	10.56	×3.68
KLASS	37.8	104.9	14.65	×2.43	42.8	102.0	5.86	×2.60	78.4	101.8	5.26	×2.53	31.7	119.5	6.16	×2.15
DAWN	38.4	57.7	26.59	×4.41	41.9	57.8	10.24	×4.55	81.3	54.9	9.42	×4.53	33.1	70.1	10.31	×3.59
CLAD	37.8	55.9	27.35	×4.54	41.6	57.0	10.19	×4.53	80.6	55.1	9.41	×4.52	32.7	70.3	10.29	×3.59
PUNT	40.9	55.3	5.60	×0.93	41.0	50.2	2.20	×0.98	79.8	55.1	1.95	×0.94	33.4	64.6	2.30	×0.80
GRID (ours)	40.2	51.1	29.31	×**4.86**	40.8	49.4	11.34	×**5.04**	81.1	49.8	10.30	×**4.95**	33.0	63.8	11.25	×**3.92**
LLaDA 2 Mini (16B)																
Top-1	77.4	426.3	1.19	×1.00	68.3	164.9	0.91	×1.00	92.3	178.3	0.84	×1.00	42.7	426.7	0.97	×1.00
EB-Sampler	74.4	131.3	3.77	×3.17	63.5	59.9	2.44	×2.69	90.8	74.3	2.06	×2.44	42.3	151.0	2.74	×2.84
Fast-dLLM	75.6	144.8	3.65	×3.07	67.5	63.3	2.34	×2.58	92.0	81.9	1.83	×2.17	42.3	162.9	2.53	×2.62
KLASS	73.8	192.6	2.51	×2.11	63.7	85.8	1.71	×1.88	91.1	98.4	1.51	×1.80	41.5	209.9	1.90	×1.96
DAWN	73.8	130.3	3.77	×3.17	64.9	56.5	2.50	×2.75	91.3	69.4	2.04	×2.41	42.3	142.8	2.75	×2.84
CLAD	75.0	137.3	3.63	×3.05	66.3	59.1	2.34	×2.58	91.4	73.5	1.86	×2.21	42.1	152.7	2.58	×2.67
PUNT	76.8	103.7	1.03	×0.87	67.1	42.2	0.65	×0.72	90.9	52.0	0.60	×0.71	42.1	112.3	0.74	×0.77
GRID (ours)	76.8	110.5	4.65	×**3.91**	65.1	48.5	2.80	×**3.09**	91.7	61.7	2.26	×**2.68**	41.9	126.1	3.10	×**3.21**
Dream-v0-Instruct-7B																
Top-1	60.4	256	3.59	×1.00	55.3	256	0.68	×1.00	78.6	256	1.42	×1.00	38.3	256	2.42	×1.00
EB-Sampler	54.3	65.6	12.87	×3.58	51.8	31.1	5.46	×8.03	78.5	59.3	6.25	×4.40	38.1	81.3	7.76	×3.21
Fast-dLLM	57.3	70.8	11.95	×3.33	54.4	33.5	5.13	×7.54	77.9	62.9	5.88	×4.14	38.2	86.9	7.24	×2.99
KLASS	58.5	108.6	6.48	×1.81	59.6	55.5	3.16	×4.65	75.0	111.3	3.09	×2.18	35.2	129.9	3.78	×1.56
DAWN	54.3	60.7	13.67	×3.81	52.9	26.8	6.21	×9.13	77.0	51.8	6.67	×4.70	37.8	71.4	8.31	×3.43
CLAD	54.3	60.6	14.59	×4.06	52.9	26.0	6.67	×9.81	76.4	48.3	7.31	×5.15	36.9	66.7	9.18	×3.79
PUNT	54.3	55.4	3.33	×0.93	56.8	23.9	1.40	×2.06	76.1	43.6	1.64	×1.15	37.2	64.4	1.94	×0.80
GRID (ours)	55.5	54.7	15.25	×**4.25**	56.5	25.0	6.71	×**9.87**	77.0	45.7	7.61	×**5.36**	37.2	61.7	9.81	×**4.05**

may not generalize to other tasks or settings. A second limitation is the loss of directionality – while inverse-ER is symmetric, counterfactual dependency is directional. This may suffice for co-decoding which is naturally symmetric, but directional information may provide richer scheduling rules and further improve performance. We leave this direction for future work. Lastly, computational scaling is another limitation, as dense ER computation requires a Laplacian pseudoinverse at each decoding step and scales as $O(|M|^3)$. For a block size of $B = 32$, used in all of our experiments, the additional cost overhead is only 0.4%, but becomes prohibitive in very large block sizes. An in-depth analysis is presented in Appendix J.

7 Conclusion

We introduced GRID, a training-free parallel sampler for MDMs that uses effective resistance on an attention-derived graph to identify weakly coupled positions for parallel decoding. The exact expansion of inverse-ER into direct and multi-hop connectivity contributions provides a structural motivation for this proxy, while our proposed benchmark, Dep-Bench, enables controlled evaluation against known graph structures and dependency depths. Across the three categories of Parallel-Bench, GRID achieves up to $\times 2$ decoding-step speedup over the fastest competing sampler, for a given quality threshold of 95% of the category-level top-1 score. Finally, across four MDMs and four leading LLM reasoning benchmarks, GRID receives the highest speedup in all 16 settings, improving over the fastest competing sampler by 7.6%-11.1% on average, while maintaining comparable accuracy. Together, these results support multi-hop connectivity as a useful signal for dependency-aware parallel decoding.

AI USE STATEMENT

In this work, we used generative AI tools solely for language polishing, improving readability, and minimal code editing for the experiments. We did not use generative AI tools to generate research ideas, develop the methodology, conduct experiments, or formulate scientific claims. We reviewed all AI-assisted text and code and verified that they accurately reflect the authors' intended meaning and experimental procedures. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

GRID is specified in Section 4 and Algorithm 1. Appendix A specifies the Dep-Bench construction, Appendix B provides the counterfactual-dependency validation, Appendix E provides the ParallelBench protocol, representative score–steps frontiers, and robustness analyses, and Appendix H provides the controlled DAWN–GRID comparisons. Appendices D, G and I provide additional implementation, hyperparameter, normalization, and ablation details. Upon acceptance, we will release the implementation code on GitHub, together with the scripts used to generate Dep-Bench and the resulting datasets.

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.

Iskander Azangulov, Teodora Pandeva, Niranjani Prasad, Javier Zazo, and Sushrut Karmalkar. Parallel sampling from masked diffusion models via conditional independence testing. In *International Conference on Learning Representations*, volume 2026, pp. 140485–140521, 2026.

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.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, et al. Llada2. 0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 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.

Shirui Chen, Jiantao Jiao, Lillian J Ratliff, and Banghua Zhu. dultra: Ultra-fast diffusion language models via reinforcement learning. *arXiv preprint arXiv:2512.21446*, 2025a.

Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. dparallel: Learnable parallel decoding for dllms. *arXiv preprint arXiv:2509.26488*, 2025b.

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. *arXiv preprint arXiv:2110.14168*, 2021.

Justin Deschenaux and Caglar Gulcehre. Beyond autoregression: Fast llms via self-distillation through time. In *International Conference on Learning Representations*, 2025.

Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Distillation of discrete diffusion through dimensional correlations. In *International Conference on Machine Learning*, 2025.

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.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjae Lee, Yuchen Zeng, Shuibai Zhang, Coleman Hooper, Yuezhou Hu, Hyung Koo, Nam Ik Cho, et al. Parallelbench: Understanding the trade-offs of parallel decoding in diffusion llms. In *International Conference on Learning Representations*, volume 2026, pp. 89900–89938, 2026.

Amin Karimi Monsefi, Nikhil Bhendawade, Manuel Rafael Ciosici, Dominic Culver, Yizhe Zhang, and Irina Belousova. Fs-dfm: Fast and accurate long text generation with few-step diffusion language models. *arXiv preprint arXiv:2509.20624*, 2025.

Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

Bumjun Kim, Dongjae Jeon, Moongyu Jeon, and Albert No. Dapd: Dependency-aware parallel decoding via attention for diffusion llms. *arXiv preprint arXiv:2603.12996*, 2026a.

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, 2026b.

D. J. Klein and M. Randić. Resistance distance. *Journal of Mathematical Chemistry*, 12(1):81–95, 1993. doi: 10.1007/BF01164627.

Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th international conference on machine learning*, volume 2002, pp. 315–322, 2002.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

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*, 2026.

Omer Luxembourg, Haim Permuter, and Eliya Nachmani. Plan for speed: Dilated scheduling for masked diffusion language models. *arXiv preprint arXiv:2506.19037*, 2025.

Mark EJ Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, 2004.

Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

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.

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

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*, 2026.

Liran Ringel, Ameen Ali, and Yaniv Romano. Dependency-guided parallel decoding in discrete diffusion language models. *arXiv preprint arXiv:2604.02560*, 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.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The diffusion duality. In *International Conference on Machine Learning*, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

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. *arXiv preprint arXiv:2505.22618*, 2025.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024.

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.

Haohui Zhang, Zhiye Wang, Xiaoying Gan, Xinbing Wang, and Bo Jiang. Leap: Unlocking dllm parallelism via lookahead early-convergence token detection. *arXiv preprint arXiv:2605.10980*, 2026.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11425–11460, 2026.

A Benchmark Datasets

All five datasets share one skeleton: three symbol-disjoint groups of 6 answers each (18 masked answers per example), with all values in 0–9. Each answer is a single-digit combination (a signed sum, or the last digit of a sum) of earlier values. The datasets vary the dependency structure, token layout, and surface form.

Below, each representative prompt is paired with its ground-truth dependency DAG. An arrow $u \to v$ denotes a *direct* dependency: v is computed using u. Given values are shown as neutral nodes. Each colored `[M]` in the prompt matches the border color of its corresponding masked node in the graph. Colors indicate correspondence only, not dependency depth. Only direct edges are drawn, so transitive dependencies appear as multi-hop paths. Each graph shows one of the three independent groups in an example. For the interleaved chain, gray lines mark prompt positions occupied by the other two groups.

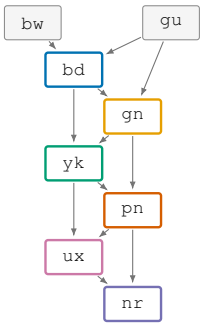
For the dependency-depth analysis in Figure 4*(b)*, masked positions with no masked predecessors are assigned depth 1, and every other masked position is assigned one plus the maximum depth of its masked predecessors.

***Arithmetic Banded Chain* (`arith_banded`).** Two seeds are given, and each answer is a signed sum of the *two immediately preceding* values. Thus every answer has two direct parents, while later answers lie progressively farther from the seeds in the dependency graph.

Example prompt

```
bw = 6
gu = 6
bd = bw - gu = [M]
gn = gu + bd = [M]
yk = bd + gn = [M]
pn = gn - yk = [M]
ux = yk + pn = [M]
nr = ux - pn = [M]
```

Ground-truth DAG



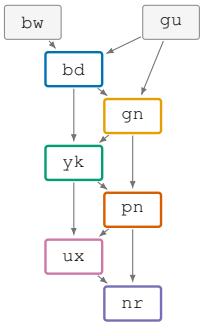
The final answers therefore depend transitively on increasingly long paths through earlier answers. The three groups are written as contiguous blocks.

***Arithmetic Interleaved Chain* (`arith_interleaved`).** This dataset uses the same within-group computation, and therefore the same ground-truth DAG, as `arith_banded`. The difference is token layout: answer lines from the three independent groups are interleaved round-robin, placing directly dependent values farther apart in the prompt.

Example prompt (one group highlighted)

```
bw = 6
gu = 6
...  seed lines from the other groups ...
bd = bw - gu = [M]
...  two other-group answer lines ...
gn = gu + bd = [M]
...  two other-group answer lines ...
yk = bd + gn = [M]
...  two other-group answer lines ...
pn = gn - yk = [M]
...  two other-group answer lines ...
ux = yk + pn = [M]
...  two other-group answer lines ...
nr = ux - pn = [M]
```

Ground-truth DAG



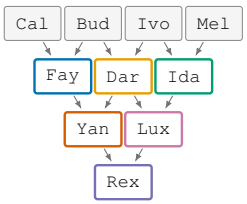
The graph is unchanged; only the positions of its nodes in the prompt are spread apart by lines from the other independent groups. This isolates token proximity from dependency structure.

***Kinship Pyramid* (`kinship_lucky`).** Four seed people are given, and each child's number is the sum of its *two named parents*. The dependency structure narrows layer by layer ($4 \to 3 \to 2 \to 1$), creating multiple routes from intermediate nodes to later answers.

Example prompt

```
Cal's is 6.  Bud's is 6.
Ivo's is 0.  Mel's is 4.
Fay, child of Cal and Bud, has lucky number [M].
Dar, child of Bud and Ivo, has lucky number [M].
Ida, child of Ivo and Mel, has lucky number [M].
Yan, child of Fay and Dar, has lucky number [M].
Lux, child of Dar and Ida, has lucky number [M].
Rex, child of Yan and Lux, has lucky number [M].
```

Ground-truth DAG

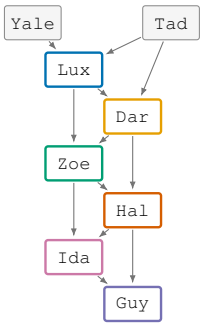


***Coins Chain* (`coins`).** This dataset preserves the same width-two DAG as `arith_banded`, but expresses its dependencies through named entities in prose rather than arithmetic equations. Each answer names the two source stalls from which it is formed.

Example prompt

```
Yale has price 6.  Tad has price 6.
Lux, formed from Yale and Tad, has price [M].
Dar, formed from Tad and Lux, has price [M].
Zoe, formed from Lux and Dar, has price [M].
Hal, formed from Dar and Zoe, has price [M].
Ida, formed from Zoe and Hal, has price [M].
Guy, formed from Hal and Ida, has price [M].
```

Ground-truth DAG



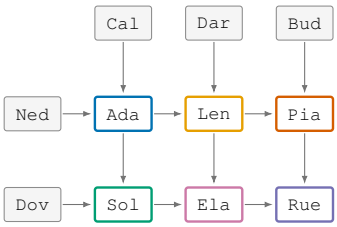
Thus `coins` and `arith_banded` share the same dependency topology while differing substantially in surface form.

***Dynasty Lattice* (`dynasty`).** A top row and a left column of seed values are given. Each interior cell is the sum of its *north and west neighbours*, yielding a two-dimensional lattice with multiple parallel routes between distant answers.

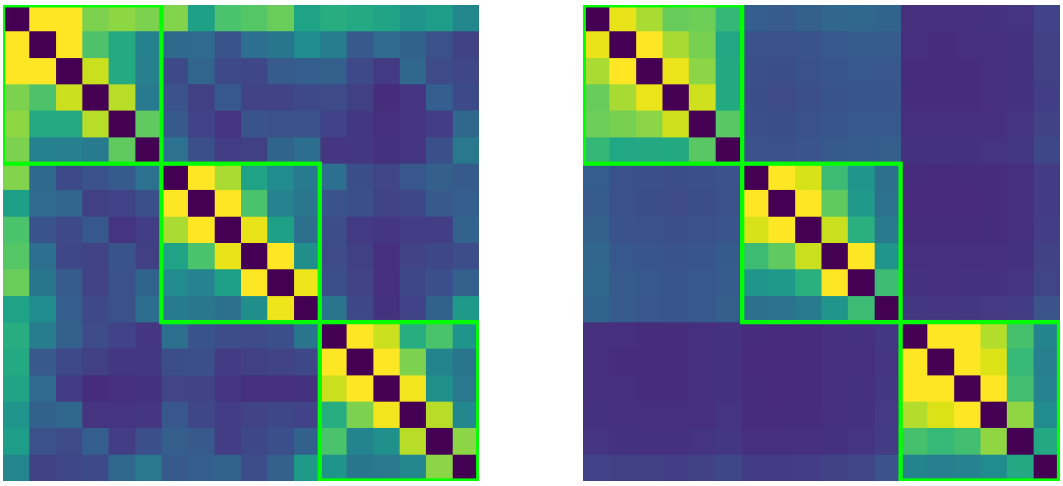
Example prompt

```
North:  Cal 6, Dar 0, Bud 4.
West:  Ned 8, Dov 7.
Ada, fed by Cal (north) and Ned (west), has yield [M].
Len, fed by Dar (north) and Ada (west), has yield [M].
Sol, fed by Ada (north) and Dov (west), has yield [M].
Pia, fed by Bud (north) and Len (west), has yield [M].
Ela, fed by Len (north) and Sol (west), has yield [M].
Rue, fed by Pia (north) and Ela (west), has yield [M].
```

Ground-truth DAG



Unlike the chain datasets, the lattice contains multiple distinct routes between distant answers, providing a controlled setting in which indirect connectivity is structurally prominent.



(a) Direct attention (b) Inverse-ER coupling

Figure 5: **Coupling maps for a real Dep-Bench example.** Arithmetic Banded Chain example using LLaDA-8B-Instruct. Green boxes mark the three independent ground-truth dependency groups. *(a)* Direct-attention coupling computed from the model's attention maps. *(b)* Inverse-ER coupling computed from GRID's modularity-denoised attention graph. Compared with the direct-attention signal, inverse ER shows broader within-group coupling by incorporating indirect paths.

B Validating the Coupling Measure Against Counterfactual KL

To complement the structural analysis of Section 3, we directly evaluate the coupling construction against the model's own counterfactual dependency. On real prompts, we compute $K_{i\leftarrow j}$ (Definition 1) by brute force and ask which coupling construction, under one fixed selection rule, selects the most mutually independent set of positions. Unlike Dep-Bench, whose dependency graph is fixed by construction, this diagnostic uses the model itself as the dependency reference. It therefore isolates the coupling construction—the ingredient that distinguishes GRID's ER-based coupling from direct-attention coupling—from the rest of the decoding procedure.

Protocol. We run up to 210 prompts per benchmark (HumanEval has only 164) with LLaDA-8B-Instruct on the same four benchmarks as Table 2 (HumanEval, MBPP, GSM8K, MATH). For each length-256 generation, we process the length-32 generation blocks in order and record at most three diagnostic states per block. At each state, we take up to 12 of the still-masked positions in the current block, uniformly subsampling 12 when more remain. For that state we build the reference dependency matrix by brute force: for each sampled position j we unmask *only* j to its argmax token and re-run the model, recording $K_{i\leftarrow j} = \mathrm{KL}\big(p_\theta(x^i \mid x_t) \,\|\, p_\theta(x^i \mid x_t^{+j})\big)$ at every other sampled position i (Definition 1, one forward pass per sampled position). Since co-decoding commits an unordered pair, the reference target is the symmetrized coupling

$$K^{\mathrm{sym}}_{ij} \;=\; \tfrac{1}{2}\big(K_{i\leftarrow j} + K_{j\leftarrow i}\big),$$

against which both measures are scored. Being built directly from K, it is available only in this diagnostic and is never used by a sampler at inference.

After recording each state, we advance it independently of the compared measures by committing the two sampled positions with the smallest row sums of K^{sym}, using their current argmax predictions. After at most three analyzed states, we proceed to the next block; positions not committed in earlier blocks remain masked. Thus, both coupling measures are evaluated on the same method-independent sequence of partially decoded states rather than on trajectories induced by either measure.

The candidate set at threshold τ is the sampled positions with confidence $\geq \tau$, swept over $\tau \in \{0.2, 0.3, \ldots, 0.9\}$. We compare two coupling scores: the *attention-edge* score, which uses the pair's direct edge, and GRID's normalized inverse-ER score q_{ij}, computed on the modularity-denoised

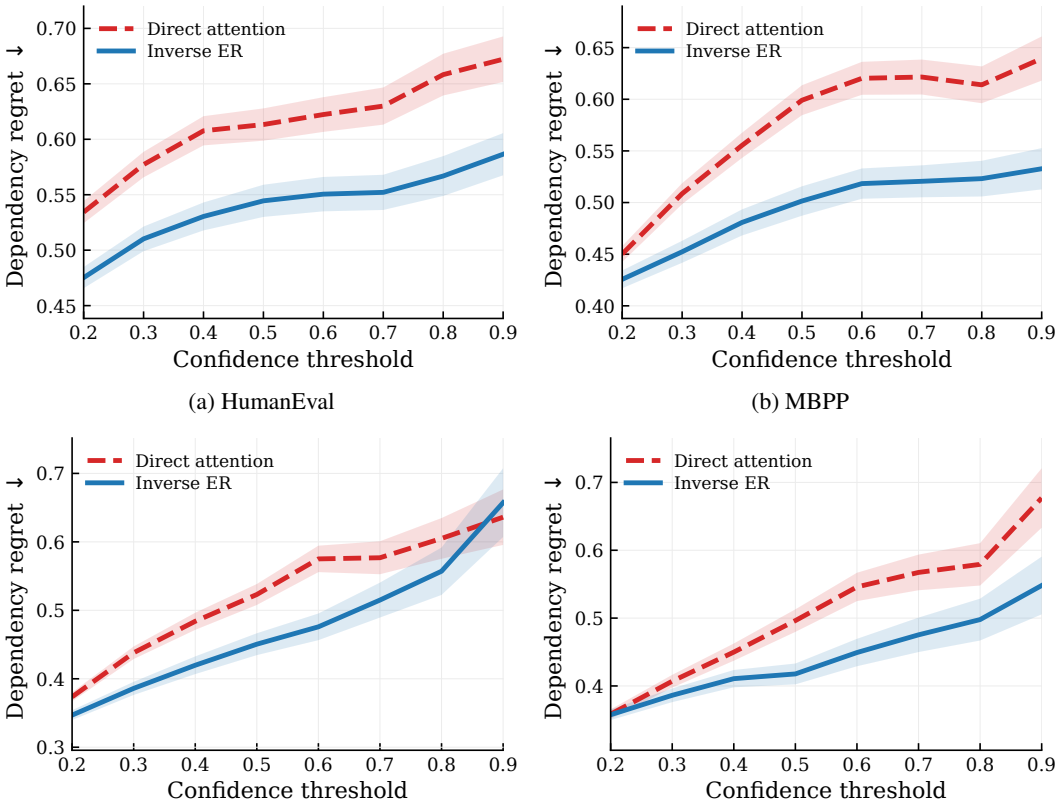
graph $\widehat{\mathbf{A}}$. For the ER computation, the graph contains all positions still masked up to the end of the current block, while selection is restricted to the sampled current-block positions. Each measure then greedily builds a size-4 set on its own score—seeded at the position of lowest total coupling, then repeatedly adding the position of smallest summed coupling to those already chosen. This is a minimum-coupling analogue of the independent-set selection in Sections 5.1 and 4; the selection rule is kept identical so that the coupling construction is the only method-dependent ingredient.

Metric. For a selected set $\mathcal{I}$, let $\mathcal{D}(\mathcal{I})$ denote the mean pairwise K^{sym} among its positions, where lower values indicate greater mutual independence. To make states comparable, we report the normalized dependency regret

$$\mathrm{Reg}(\mathcal{I}) = \frac{\mathcal{D}(\mathcal{I}) - \mathcal{D}_\star}{\mathcal{D}_{\mathrm{avg}} - \mathcal{D}_\star},$$

where $\mathcal{D}_\star$ is the minimum over all candidate k-subsets and $\mathcal{D}_{\mathrm{avg}}$ is their mean. With at most 12 sampled positions and $k = 4$, there are at most $\binom{12}{4} = 495$ subsets per state, so both quantities are computed exhaustively. Thus, 0 is the best-possible k-subset, while 1 corresponds to the mean for a uniformly random k-subset. Lower is better. A state contributes to threshold τ only when at least $k{+}1 = 5$ of its sampled positions clear τ, ensuring a nontrivial choice among k-subsets. Stricter thresholds leave fewer qualifying states, from $\sim$3600 states at the loosest threshold to $\sim$200 at the strictest.

Result. Across all four datasets and all eight thresholds (Figure 6), inverse-ER coupling yields lower normalized regret than direct-attention coupling at every point—the gap is 0.01–0.11 in normalized units (mean $\approx$0.07). Because the two curves use the same reference K^{sym}, candidate sets, and selection rule and differ only in the coupling construction, the comparison isolates that construction from the rest of the sampler. $\mathcal{D}_\star$ is computed by exhaustively evaluating all candidate subsets using K^{sym}, and both measures remain above zero regret in the reported results. The relevant result is their consistent ordering: inverse ER has lower regret at every threshold, consistent with the multi-hop motivation of Section 3 and with Dep-Bench (Section 5.1) under a distinct model-grounded reference. Thus, the advantage appears at the level of the coupling construction, independently of the full sampler.



(a) HumanEval (b) MBPP

(c) GSM8K (d) MATH

Figure 6: Normalized dependency regret vs. confidence threshold τ on four benchmarks (LLaDA-8B-Instruct). Lower is better. 0 corresponds to the lowest-dependency candidate k-subset, while 1 corresponds to the mean of a uniformly random candidate k-subset. Each panel compares direct-attention and inverse-ER coupling under the same minimum-coupling selection rule. Shaded bands denote $\pm$SEM across diagnostic states. Inverse-ER coupling yields lower regret at every threshold on all four benchmarks.

C Derivation of the Effective-Resistance Expansion

We derive Equation (1). Consider a connected undirected weighted graph on vertex set V with symmetric, nonnegative adjacency matrix $\mathbf{A}$ and $A_{uu} = 0$. Fix distinct nodes i and j, let $T = \{i, j\}$ and $S = V \setminus T$, and define the full weighted degrees $d_u = \sum_{v \in V} A_{uv}$. Let $\mathbf{D} = \mathrm{diag}(d_u)$ and $\mathcal{L} = \mathbf{D} - \mathbf{A}$.

Partition the Laplacian according to the terminal vertices T and the intermediate vertices S:

$$\mathcal{L} = \begin{bmatrix} \mathcal{L}_{TT} & \mathcal{L}_{TS} \\ \mathcal{L}_{ST} & \mathcal{L}_{SS} \end{bmatrix}.$$

Since the graph is connected and T is nonempty, $\mathcal{L}_{SS}$ is nonsingular. Eliminating the intermediate vertices gives the two-terminal Schur complement

$$\mathcal{L}_{\mathrm{red}} = \mathcal{L}_{TT} - \mathcal{L}_{TS}\mathcal{L}_{SS}^{-1}\mathcal{L}_{ST}.$$

This reduction preserves the effective resistance between i and j. The reduced two-node graph therefore has edge weight R_{ij}^{-1}, so its Laplacian is

$$\mathcal{L}_{\mathrm{red}} = R_{ij}^{-1} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Using $\mathcal{L}_{ij} = -A_{ij}$, $\mathcal{L}_{iS} = -\mathbf{A}_{iS}$, and $\mathcal{L}_{Sj} = -\mathbf{A}_{Sj}$, the off-diagonal entry gives

$$R_{ij}^{-1} = A_{ij} + \mathbf{A}_{iS}\mathcal{L}_{SS}^{-1}\mathbf{A}_{Sj}. \tag{5}$$

Let $\mathbf{D}_S = \mathrm{diag}(d_k)_{k\in S}$, where each d_k is the full weighted degree in the original graph. Then

$$\mathcal{L}_{SS} = \mathbf{D}_S - \mathbf{A}_{SS} = \mathbf{D}_S \left(\mathbf{I} - \mathbf{D}_S^{-1}\mathbf{A}_{SS}\right).$$

Define

$$\mathbf{Q} = \mathbf{D}_S^{-1}\mathbf{A}_{SS}.$$

Since $\mathbf{Q} = \mathbf{D}_S^{-1}\mathbf{A}_{SS}$, it is substochastic. Because the graph is connected, every node in S can reach the terminal set T, which implies $\rho(\mathbf{Q}) < 1$. Hence

$$(\mathbf{I} - \mathbf{Q})^{-1} = \sum_{m=0}^{\infty} \mathbf{Q}^m,$$

and therefore

$$\mathcal{L}_{SS}^{-1} = \sum_{m=0}^{\infty} \left(\mathbf{D}_S^{-1}\mathbf{A}_{SS}\right)^m \mathbf{D}_S^{-1}.$$

Substituting into Equation (5) gives

$$\begin{aligned} R_{ij}^{-1} = A_{ij} &+ \mathbf{A}_{iS}\mathbf{D}_S^{-1}\mathbf{A}_{Sj} \\ &+ \mathbf{A}_{iS}\mathbf{D}_S^{-1}\mathbf{A}_{SS}\mathbf{D}_S^{-1}\mathbf{A}_{Sj} + \cdots, \end{aligned}$$

or, entrywise,

$$R_{ij}^{-1} = A_{ij} + \sum_{k\in S} \frac{A_{ik}A_{kj}}{d_k} + \sum_{k,\ell\in S} \frac{A_{ik}A_{k\ell}A_{\ell j}}{d_k d_\ell} + \cdots,$$

which is Equation (1). The successive terms correspond to one-hop, two-hop, three-hop, and longer multi-hop contributions. More precisely, they sum weighted walks from i to j whose intermediate vertices lie in S. Intermediate vertices may repeat at higher orders.

The derivation above assumes a connected graph. When the denoised graph used in practice is disconnected, effective resistance is computed separately within each connected component. Pairs in different components are assigned zero inverse-ER coupling, equivalently $R_{ij} = \infty$, as described in Appendix D.

D Implementation Details

Fixed as part of the method. GRID always applies modularity denoising to the attention graph, computes effective resistance on the denoised graph, and forms the normalized inverse-ER coupling score $q_{ij} = R_{ij}^{-1}/\bar{d}$. This normalization removes uniform graph-scale drift as the masked graph shrinks, so the scheduled threshold $q_t^\star$ retains a consistent meaning across decoding steps. Positions are selected by the greedy independent-set rule of Algorithm 1. These components are fixed as part of the method rather than tuned.

Restricting the graph to masked positions. GRID needs coupling scores only among the currently masked positions. We therefore compute resistance on the graph induced by M rather than retaining prompt tokens and already-unmasked positions as intermediate resistance vertices, which would substantially enlarge the graph even though they cannot be selected. These positions still shape the attention weights through the full model forward pass before this restriction. Thus, indirect routes in the resistance computation run through masked positions only. This restriction is a practical approximation that keeps the graph small.

Disconnected denoised graphs. The denoising in Equation (2) can set edge weights to zero and disconnect $\widehat{\mathbf{A}}$. We therefore compute effective resistance separately within each connected component. For positions in different components, we define $q_{ij} = 0$, equivalently treating $R_{ij} = \infty$, since no positive-weight path connects them in the denoised graph.

Generation settings. We use block length 32 for all models. The generation length is 256 for LLaDA-8B-Instruct, LLaDA-1.5, and Dream-v0-Instruct-7B, and 512 for LLaDA 2 Mini, following its standard decoding setup. For all LLaDA 2 Mini runs, decoding stops once a fully generated answer ending in EOS is obtained, rather than continuing to decode tokens that would subsequently be discarded. Since evaluation already truncates the generated output at the first EOS, this does not change the scored answer; it only avoids unnecessary computation after generation is complete. Consequently, the reported number of decoding steps for LLaDA 2 Mini can be smaller than the maximum generation length even for top-1 decoding (e.g., 426.3 steps on HumanEval in Table 2). This early stopping is applied consistently to all LLaDA 2 Mini methods reported in Table 2 and is not used for the other models.

Shared GRID settings. The coupling-threshold schedule follows eq. (4). The attention-sink threshold τ_{sink} is not tuned but taken from the DAWN implementation: 0.01 for the LLaDA models and 0.03 for Dream-v0-Instruct-7B. For LLaDA-8B-Instruct, LLaDA-1.5, and Dream-v0-Instruct-7B, we use $\tau_{\text{high}} = 0.9$; for LLaDA 2 Mini, τ_{high} is included in the HumanEval grid search described below.

Tuned GRID hyperparameters. Following DAWN, we select GRID's hyperparameters by grid search on HumanEval, separately for each model, and reuse the selected configuration unchanged on the other three benchmarks. Among the evaluated configurations, we select the fastest operating point whose HumanEval accuracy is at least that of DAWN under the same model. We use DAWN as the reference because it is the closest structural baseline to GRID. For LLaDA-8B-Instruct, LLaDA-1.5, and Dream-v0-Instruct-7B, we search the base coupling threshold $q_0 \in \{0.01, 0.03, 0.05, 0.07, 0.09\}$, the confidence floor $\tau_{\text{low}} \in \{0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$, and the schedule rate $\delta \in \{0.01, 0.03, 0.05, 0.07, 0.09, 0.11, 0.15, 0.2\}$. The selected configurations $(q_0, \tau_{\text{low}}, \delta)$ are $(0.01, 0.25, 0.15)$ for LLaDA-8B-Instruct, $(0.07, 0.5, 0.03)$ for LLaDA-1.5, and $(0.01, 0.35, 0.07)$ for Dream-v0-Instruct-7B.

For LLaDA 2 Mini, we search $q_0 \in \{0.01, 0.03, 0.05, 0.07, 0.09\}$, $\tau_{\text{low}} \in \{0.3, 0.4, 0.5, 0.6\}$, $\delta \in \{0.05, 0.1, 0.15, 0.2\}$, and $\tau_{\text{high}} \in \{0.9, 0.93, 0.95, 0.97\}$. The selected LLaDA 2 Mini configuration $(q_0, \tau_{\text{low}}, \delta, \tau_{\text{high}})$ is $(0.03, 0.6, 0.1, 0.97)$. A one-at-a-time sensitivity analysis of the LLaDA-1.5 configuration is provided in Appendix I.3.

Baselines. For KLASS, DAWN, and CLAD, we use the official implementations released by the respective authors. For EB-Sampler and Fast-dLLM, we use the implementations provided in the official DAWN codebase. Since no official implementation of PUNT is publicly available, we implement it following the method described in the paper.

For LLaDA-8B-Instruct, LLaDA-1.5, and Dream-v0-Instruct-7B, the baselines use their published settings. Since the competing samplers do not report settings for LLaDA 2 Mini, for this model we tune their applicable hyperparameters by grid search on HumanEval and reuse each selected configuration unchanged on MBPP, GSM8K, and MATH. Thus, both GRID and the tuned baselines use HumanEval for model-specific hyperparameter selection.

For DAWN on LLaDA 2 Mini, we use $\tau_{\text{sink}} = 0.01$ and select $\tau_{\text{edge}} = 0.05$ from $\{0.03, 0.05, 0.07, 0.09, 0.11\}$, $\tau_{\text{induce}} = 0.85$ from $\{0.7, 0.75, 0.8, 0.85, 0.9\}$, $\tau_{\text{low}} = 0.9$ from $\{0.75, 0.8, 0.85, 0.9, 0.95\}$, and $\tau_{\text{high}} = 0.95$ from $\{0.9, 0.93, 0.95, 0.97\}$. For EB-Sampler, we use `factor`$= 0.3$, selected from $\{1, 0.7, 0.5, 0.3, 0.1\}$. For Fast-dLLM, we use `threshold`$= 0.95$, selected from $\{0.7, 0.75, 0.8, 0.85, 0.9, 0.95\}$. For CLAD, we use `threshold`$= 0.9$, selected from $\{0.7, 0.75, 0.8, 0.85, 0.9, 0.95\}$. For KLASS and PUNT, we use the same default settings as for the other LLaDA models.

The threshold-sweep experiments are separate controlled experiments that run the compared methods outside their published operating points (Appendix H) and degree-normalize the dependency scores before thresholding (Appendix G).

Hardware. All experiments use NVIDIA A10G GPUs with 24 GB of memory per GPU. LLaDA-8B-Instruct, LLaDA-1.5, and Dream-v0-Instruct-7B are run on a single GPU, while LLaDA 2 Mini is loaded across two GPUs to accommodate the larger model. All methods compared within a given model use the same hardware configuration. Tokens-per-second figures are therefore directly com-

parable across samplers for the same model, but not directly across models using different hardware configurations or to measurements on other accelerators. The threshold sweeps, controlled analyses, and per-step cost measurements of Figure 14 use LLaDA-8B-Instruct on a single GPU running without contention.

E PARALLELBENCH EVALUATION

ParallelBench (Kang et al., 2026) contains 17 tasks grouped into three categories: Waiting Line, Text Writing, and Puzzles. We merge the tasks within each category for category-level evaluation and evaluate LLaDA-8B-Instruct on all three categories. Each task retains its benchmark-specific score, and the category-level score and average decoding steps are computed by averaging over the constituent tasks as defined below. Waiting Line uses generation and block length 32, while Text Writing and Puzzles use generation and block length 64. Top-1 decoding, which decodes one position per step, provides the reference score for each task.

Parameter sweeps. For each parallel sampler, we evaluate every parameter setting on every task and aggregate the results within each category. Operating-point selection is performed only after this aggregation, so the same parameter setting is used across all tasks in a category. For GRID, we fix $\delta = 0.15$ to the value selected for LLaDA-8B-Instruct on HumanEval and sweep

$$q_0 \in \{0.01, 0.03, 0.05, 0.07, 0.10, 0.15, 0.20\},$$

$$\tau_{\text{low}} \in \{0.25, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.85, 0.90\}.$$

For DAWN, we sweep

$$\tau_{\text{edge}} \in \{0.01, 0.05, 0.10, 0.15, 0.20, 0.30, 0.40\},$$

$$\tau_{\text{low}} \in \{0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.85, 0.90\}.$$

For Fast-dLLM, we sweep

$$\texttt{threshold} \in \{0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 0.99\}.$$

For EB-Sampler, we sweep

$$\texttt{factor} \in \{0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.00, 1.20, 1.50, 2.00\}.$$

For KLASS, we fix the confidence threshold to 0.6 and history length to 2 and sweep

$$\texttt{kl_threshold} \in \{0.01, 0.05, 0.10, 0.20, 0.30, 0.50, 0.70, 0.90, 1.20, 1.50, 2.00, 3.00, 5.00, 10.00\}.$$

For CLAD, we use max symmetrization and sweep

$$\texttt{clad_threshold} \in \{0.30, 0.40, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 0.97\}.$$

PUNT is omitted because it performs additional model evaluations within a nominal decoding step, making nominal step counts non-comparable.

Category-level matched-quality metric. Let $\mathcal{C}$ denote a ParallelBench category, $\theta \in \Theta_m$ an evaluated parameter setting of method m, and $t \in \mathcal{C}$ a constituent task. The same setting θ is used across all tasks in $\mathcal{C}$. We define the category-level score and average decoding steps as

$$Q(m, \mathcal{C}; \theta) = \frac{1}{|\mathcal{C}|} \sum_{t \in \mathcal{C}} Q(m, t; \theta), \qquad S(m, \mathcal{C}; \theta) = \frac{1}{|\mathcal{C}|} \sum_{t \in \mathcal{C}} \text{Steps}(m, t; \theta),$$

where $Q(m, t; \theta)$ is the benchmark-specific score on task t. Each constituent task therefore contributes equally to the category-level evaluation. The corresponding category-level top-1 reference score is

$$Q_{\text{top1}}(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{t \in \mathcal{C}} Q_{\text{top1}}(t).$$

For quality target $\alpha \in \{0.90, 0.95, 0.99\}$, every sampler is required to satisfy the same category-level quality constraint. Its matched-quality step count is

$$S_\alpha(m, \mathcal{C}) = \min_{\theta \in \Theta_m} \{S(m, \mathcal{C}; \theta) : Q(m, \mathcal{C}; \theta) \geq \alpha Q_{\text{top1}}(\mathcal{C})\}.$$

If no evaluated setting reaches the target, we set $S_\alpha(m, \mathcal{C}) = +\infty$.

Let

$$\mathcal{B} = \{\text{Fast-dLLM, EB-Sampler, KLASS, DAWN, CLAD}\}.$$

The fastest competing sampler is selected once for each category and quality target:

$$S_\alpha^{\text{base}}(\mathcal{C}) = \min_{m \in \mathcal{B}} S_\alpha(m, \mathcal{C}).$$

We report GRID's matched-quality decoding-step speedup as

$$r_{\alpha,\mathcal{C}} = \frac{S_\alpha^{\text{base}}(\mathcal{C})}{S_\alpha(\text{GRID}, \mathcal{C})}.$$

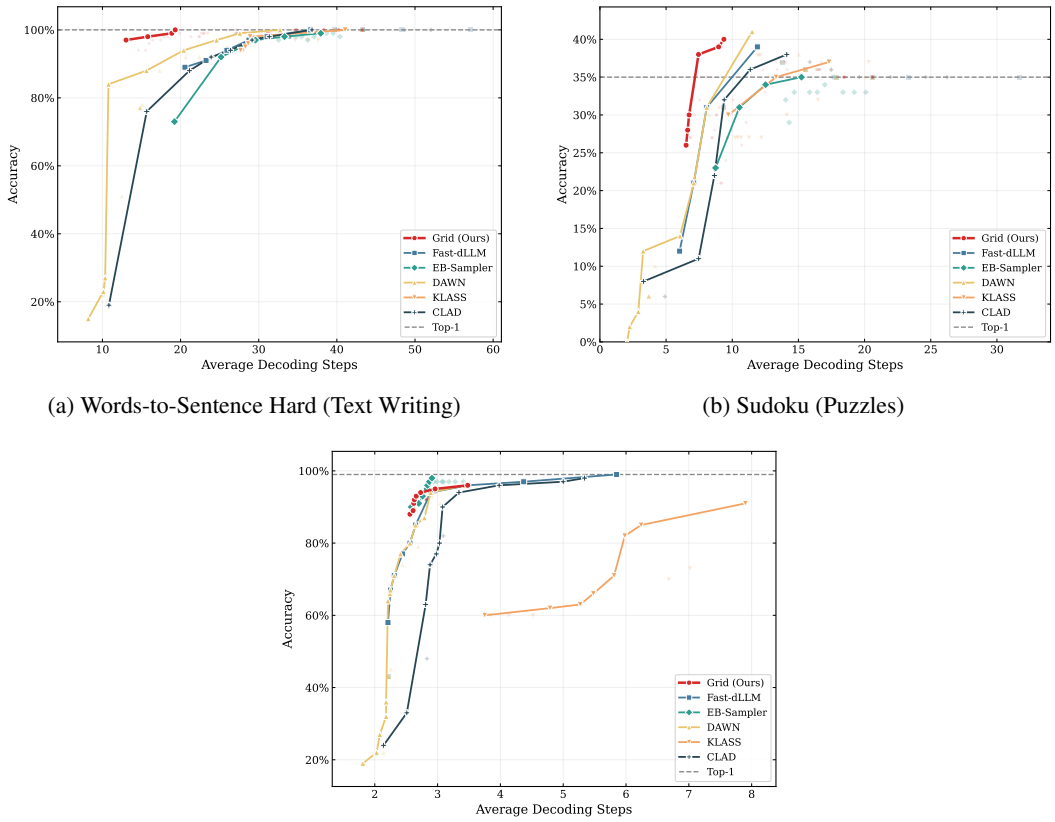
Thus, $r_{\alpha,\mathcal{C}} > 1$ means that GRID requires fewer decoding steps than the fastest competing sampler, $r_{\alpha,\mathcal{C}} = 1$ denotes parity, and $r_{\alpha,\mathcal{C}} < 1$ means that the competing sampler requires fewer steps.

Representative score–steps frontiers. Figure 7 shows one representative task from each ParallelBench category: Sudoku from Puzzles, Words-to-Sentence Hard from Text Writing, and Reverse from Waiting Line. These per-task plots provide qualitative views of the score–steps trade-off, while the summary results in Table 1 use the category-level evaluation defined above. The plots show both the complete set of evaluated operating points and the corresponding Pareto frontiers.

Robustness. To evaluate sensitivity to GRID's schedule parameter, we fix δ in turn to each tested value

$$\delta \in \{0.01, 0.03, 0.05, 0.07, 0.09, 0.11, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40\}$$

and recompute the category-level matched-quality metric while sweeping (q_0, τ_{low}) as above. At the 90% quality target, GRID's decoding-step speedup over the fastest competing sampler ranges from $\times 1.01$–$\times 1.05$ on Waiting Line, $\times 1.39$–$\times 1.76$ on Text Writing, and $\times 1.30$–$\times 1.65$ on Puzzles. At the 95% target, the corresponding ranges are $\times 1.06$–$\times 1.08$, $\times 1.79$–$\times 2.10$, and $\times 1.30$–$\times 1.64$, respectively, across the evaluated values of δ.



(a) Words-to-Sentence Hard (Text Writing)

(b) Sudoku (Puzzles)

(c) Reverse (Waiting Line)

Figure 7: **Representative ParallelBench score–steps trade-offs.** Faint markers show all evaluated operating points from the parameter sweeps, while solid curves show the corresponding Pareto frontiers. The dashed line denotes the top-1 task score. We show Sudoku as a representative Puzzles task, Words-to-Sentence Hard as a representative Text Writing task, and Reverse as a representative Waiting Line task.

F Additional DAPD Evaluation

We additionally evaluate DAPD (Kim et al., 2026a) using its official implementation on LLaDA-8B-Instruct. Table 3 reports the resulting operating points under our benchmark evaluation framework.

DAPD ranks masked positions using a score that combines prediction confidence with the node's degree in the attention graph, and greedily selects nonconflicting positions. In our evaluation, we observed that positions with relatively low prediction confidence can nevertheless receive high ranking scores when their attention-graph degree is high, causing them to be prioritized for decoding. We believe that committing such low-confidence positions contributes to the observed accuracy degradation despite the reduced decoding-step count. We therefore report DAPD separately from the main E2E comparison.

G Degree Normalization of Dependency Scores

Attention is computed over the full sequence, whereas the dependency graph used by GRID contains only the currently masked positions. As positions are committed, this induced graph changes in size and the overall magnitude of its retained edge weights can change. A fixed threshold on raw inverse ER could therefore drift in meaning because of global graph scale rather than relative connectivity.

We remove this multiplicative scale dependence using the mean weighted degree $\bar{d} = \mathrm{vol}(\widehat{\mathbf{A}})/|M|$ of the current denoised graph. Consider a uniform rescaling $\mathbf{A} \mapsto s\mathbf{A}$ with $s > 0$. Then $d_i \mapsto sd_i$ and $\mathrm{vol}(\mathbf{A}) \mapsto s\mathrm{vol}(\mathbf{A})$, so Equation (2) gives $\widehat{\mathbf{A}} \mapsto s\widehat{\mathbf{A}}$. Effective resistance consequently scales as $R_{ij} \mapsto R_{ij}/s$, while $\bar{d} \mapsto s\bar{d}$. Hence $q_{ij} = R_{ij}^{-1}/\bar{d}$ is invariant to uniform edge-weight rescaling.

Within a decoding step, $\bar{d}$ is common to all pairs, so the normalization preserves their ordering. It removes only the global multiplicative scale; changes in relative edge weights or graph structure remain reflected in q_{ij}. For the attention-edge score, the corresponding scale-invariant quantity is $A_{ij}/\bar{d}$.

For GRID, the normalization is part of the method (Section 4). For the attention-edge score, we apply the analogous normalization only in controlled experiments that sweep a dependency threshold to trace a frontier (Figures 4 and 8 to 10). This isolates the quality of the dependency score from changes in graph scale. The E2E comparison in Table 2 uses DAWN exactly as published, without this additional normalization.

H Controlled DAWN–GRID Comparisons

This appendix details the controlled DAWN–GRID comparisons used in Section 5.1 and on the four standard code and math benchmarks. Both comparisons use a shared selection procedure to isolate the dependency-scoring mechanism.

Shared simplified protocol. We use simplified DAWN and GRID variants with the same confidence-ordered greedy selection procedure while retaining their respective dependency scores. We disable the immediate-commit stage by setting $\tau_{\mathrm{high}} = 1.1$, remove the confidence floor with $\tau_{\mathrm{low}} = 0$, and disable DAWN's anchor-guided stage with $\tau_{\mathrm{induce}} = 1.1$. Both variants use $\tau_{\mathrm{sink}} = 0.01$. GRID retains its inverse-ER dependency score, while DAWN retains its direct attention-edge score. Both scores are degree-normalized before thresholding, as detailed in Appendix G. These simplified variants are used only for controlled analysis.

Dep-Bench. On Dep-Bench, each variant sweeps its dependency-score threshold to trace the slot-accuracy–decoding-steps frontier in Figure 4*(a)*. For Figure 4*(b)*, we compare operating points at approximately 9 average decoding steps. Figure 4*(c)* evaluates each threshold setting relative to the known task-dependency graph.

Standard code and math benchmarks. We repeat the same controlled comparison with LLaDA-8B-Instruct on HumanEval, MBPP, GSM8K, and MATH, using task accuracy in place of slot accuracy (Figure 8). All runs use generation length 256 and block length 32. GRID sweeps its inverse-ER

Table 3: DAPD results on LLaDA-8B-Instruct under our benchmark evaluation. ΔAcc is relative to top-1 decoding.

Benchmark	Acc. ↑	ΔAcc	Steps ↓	TPS ↑
HumanEval	31.1	−11.0	53.5	27.86
MBPP	38.9	−3.1	70.7	6.08
GSM8K	52.5	−26.2	55.9	8.82
MATH	23.2	−9.7	64.1	10.80

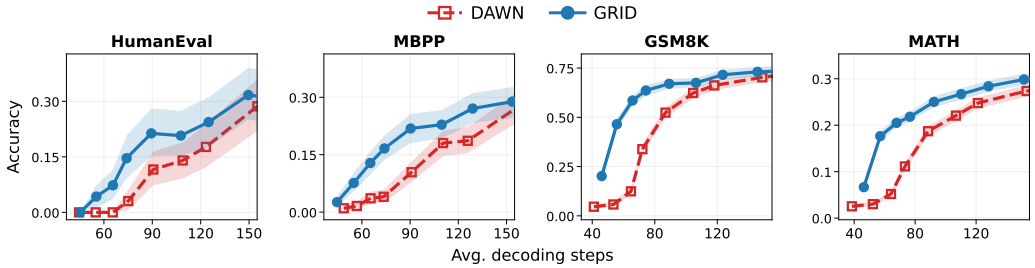


Figure 8: **Controlled comparison of simplified DAWN and GRID variants on standard code and math benchmarks.** Accuracy vs. average decoding steps with LLaDA-8B-Instruct. Both variants use the same confidence-ordered greedy selection procedure while retaining their respective dependency scores. Shaded bands are 95% bootstrap confidence intervals.

threshold over $[0.06, 0.46]$, while DAWN sweeps its direct-edge threshold τ_{edge} over $[0.0057, 0.162]$. The ten threshold values are paired one-to-one and calibrated on HumanEval to matched average step counts spanning approximately 45 to 200 steps. The remaining benchmarks land near, but not exactly on, these targets.

Across all four benchmarks, GRID is at least as accurate as DAWN over the evaluated decoding budgets, with the advantage increasing at tighter budgets. On GSM8K at around 65 average steps, for example, GRID reaches 0.59 accuracy versus 0.12 for DAWN.

I Ablations

I.1 Denoising Without Effective Resistance

To isolate whether modularity denoising alone accounts for the improvement, we add a third curve—*modularity-normed attention*—which uses $\widehat{A}_{ij}$ (Equation (2)) directly as the coupling score, bypassing effective resistance. We evaluate it under the same controlled protocol as the direct-edge attention and inverse-ER variants on both Dep-Bench (Section 5.1) and the four standard code and math benchmarks (Appendix H). All other settings are identical within each comparison.

In both settings (Figures 9 and 10), the modularity-normed variant is confined to a narrow range of decoding steps (≈5–7 on Dep-Bench and ≈50–80 on the standard code and math benchmarks) across the threshold sweep. Subtracting the degree-driven null sets many pairwise coupling scores to zero, leaving a sparse conflict graph. These zero-score pairs remain nonconflicting across the sweep, so the greedy independent set admits many positions per step and the variant remains in a low-step regime. Empirically, varying the threshold changes the selected-set size only modestly, resulting in the narrow range of decoding steps. Effective resistance dominates both alternatives across the evaluated step budgets, showing that the improvement comes from the ER connectivity measure rather than modularity denoising alone.

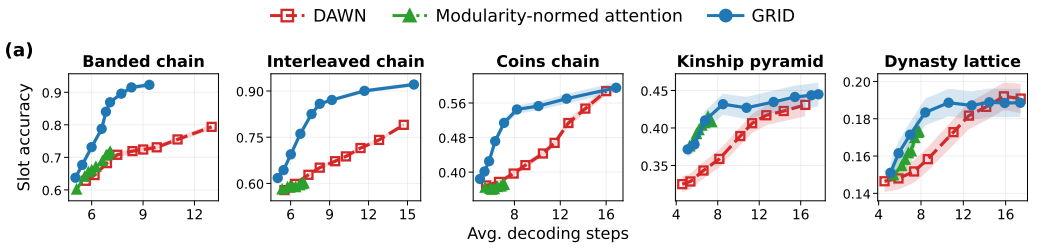


Figure 9: Ablation on Dep-Bench (Section 5.1, LLaDA-8B-Instruct, 1500 examples per dataset). Slot accuracy vs. average decoding steps (up-left is better). The modularity-normed variant is confined to ≈5–7 steps. Shaded bands are 95% bootstrap confidence intervals.

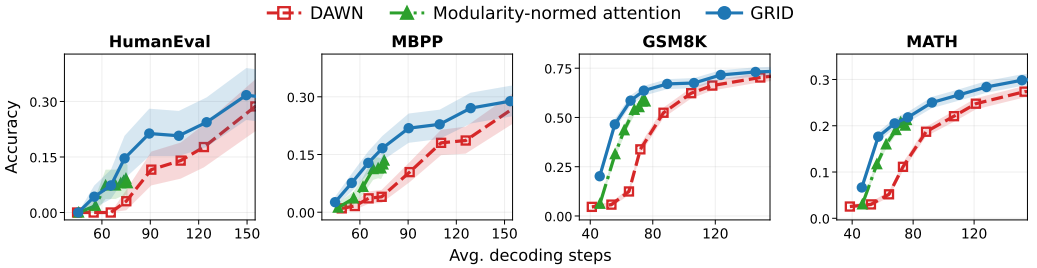


Figure 10: Ablation on the four standard code and math benchmarks (LLaDA-8B-Instruct, controlled protocol of Appendix H). The same three coupling measures are compared, and the modularity-normed variant is confined to ≈50–80 steps. Shaded bands are 95% bootstrap confidence intervals.

I.2 Alternative Multi-Hop Scores

Here we test whether the benefit of modeling multi-hop connectivity extends beyond effective resistance. Under the same controlled Dep-Bench protocol described in Appendix H, we compare inverse ER with two standard alternative multi-hop graph measures, Katz similarity (Katz, 1953) and the heat kernel (Kondor & Lafferty, 2002). The ER, Katz, and heat-kernel variants use the same modularity-denoised attention graph, confidence-ordered greedy selection, and disabled confidence gates, so they differ only in the pairwise multi-hop coupling score. We additionally include direct-edge attention as the one-hop reference.

Let $\widehat{\mathbf{A}}$ be the modularity-denoised adjacency and define $\widetilde{\mathbf{A}} = \widehat{\mathbf{A}}/\bar{d}$, where $\bar{d}$ is its mean weighted degree. For Katz similarity, let $\rho = \lambda_{\max}(\widetilde{\mathbf{A}})$ and define

$$\mathbf{K}_\gamma = \left(\mathbf{I} - \frac{\gamma}{\rho}\widetilde{\mathbf{A}}\right)^{-1} - \mathbf{I} = \sum_{m\geq 1}\left(\frac{\gamma}{\rho}\right)^m \widetilde{\mathbf{A}}^{\,m}.$$

The spectral-radius normalization makes $\gamma \in (0,1)$ a scale-independent range parameter and guarantees convergence. We evaluate $\gamma \in \{0.3, 0.6, 0.9\}$.

For the heat kernel, let

$$\widetilde{\mathcal{L}} = \widetilde{\mathbf{D}} - \widetilde{\mathbf{A}}$$

be the graph Laplacian of $\widetilde{\mathbf{A}}$ and define

$$\mathbf{H}_t = \exp(-t\widetilde{\mathcal{L}}).$$

The diffusion time t controls the effective range of the score, and we evaluate $t \in \{1, 2, 4\}$. In both cases, larger off-diagonal scores indicate stronger coupling. For every value of γ or t, we independently sweep the coupling threshold to trace the slot-accuracy–decoding-steps frontier. Because the score scale changes with the range parameter, each value uses a six-point threshold grid chosen to

span the relevant decoding-step range, and each operating point is plotted at its measured average number of steps.

Figures 11 and 12 show that the multi-hop benefit is not specific to effective resistance. On the chain benchmarks, suitable Katz and heat-kernel settings can recover substantial gains over direct-edge attention and preserve accuracy better as dependency depth increases. Performance varies with γ and t, and the relative ordering also varies across dependency structures. Thus, alternative multi-hop measures can also provide useful dependency scores. ER provides a competitive choice without requiring an additional decay or diffusion-time parameter to determine the multi-hop range.

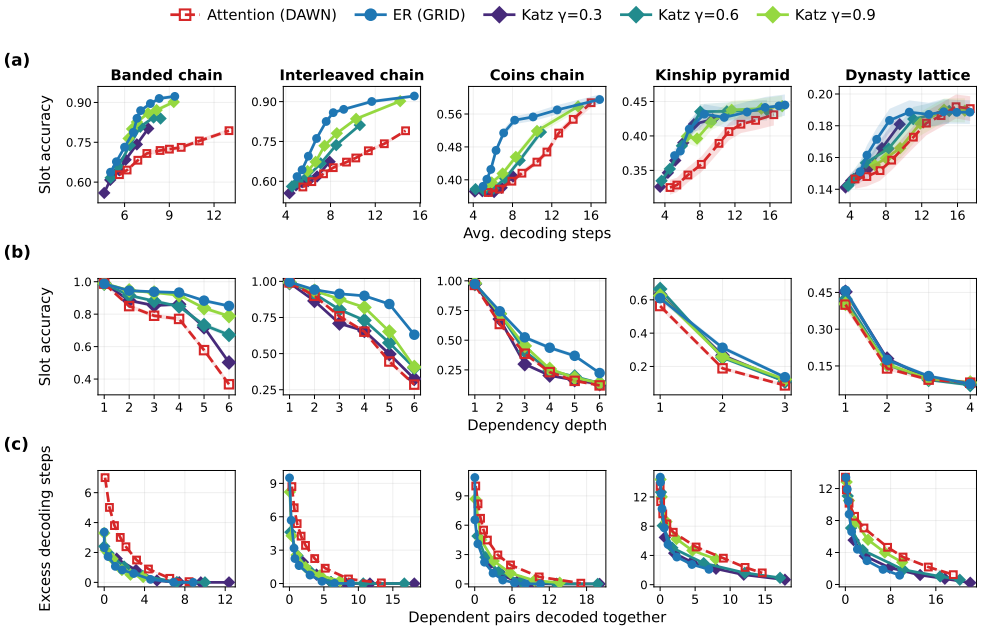


Figure 11: Katz multi-hop score ablation on Dep-Bench (LLaDA-8B-Instruct). Each Katz curve fixes $\gamma \in \{0.3, 0.6, 0.9\}$ and sweeps its coupling threshold. *(a)* Slot accuracy vs. average decoding steps. *(b)* Accuracy by dependency depth, using for each score and dataset the swept operating point closest to 9 average decoding steps. *(c)* Excess decoding steps vs. dependent pairs decoded together. Attention and ER are the same reference variants as in Figure 4.

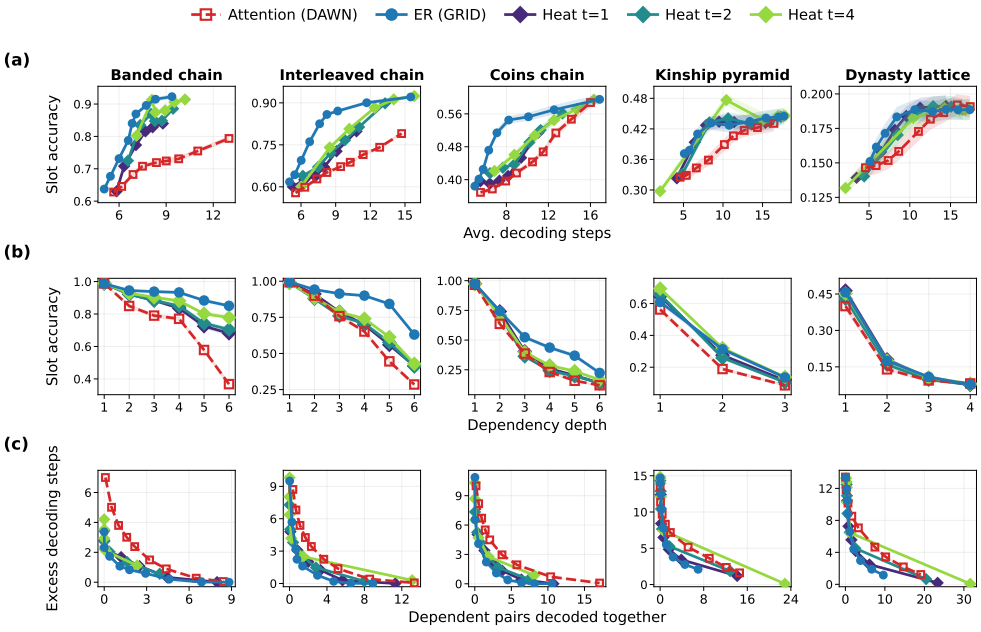


Figure 12: Heat-kernel multi-hop score ablation on Dep-Bench (LLaDA-8B-Instruct). Each heat-kernel curve fixes $t \in \{1, 2, 4\}$ and sweeps its coupling threshold. Panels and reference variants follow Figure 11.

I.3 Hyperparameter Sensitivity

We evaluate one-at-a-time sensitivity of GRID's three tuned hyperparameters on HumanEval with LLaDA-1.5. Starting from the reported configuration $(q_0, \tau_{\text{low}}, \delta) = (0.07, 0.50, 0.03)$, we vary one parameter while fixing the other two. Figure 13 reports settings spanning $q_0 \in [0.03, 0.13]$, $\tau_{\text{low}} \in [0.25, 0.80]$, and $\delta \in [0.01, 0.30]$. These diagnostic ranges extend beyond the original tuning grid and are not used to change the reported configuration.

Increasing q_0 or δ makes the coupling criterion more permissive and generally reduces the number of decoding steps, while lowering τ_{low} admits more candidates for parallel decoding. More aggressive settings eventually trade accuracy for fewer steps, while several nearby settings achieve similar accuracy to the reported configuration.

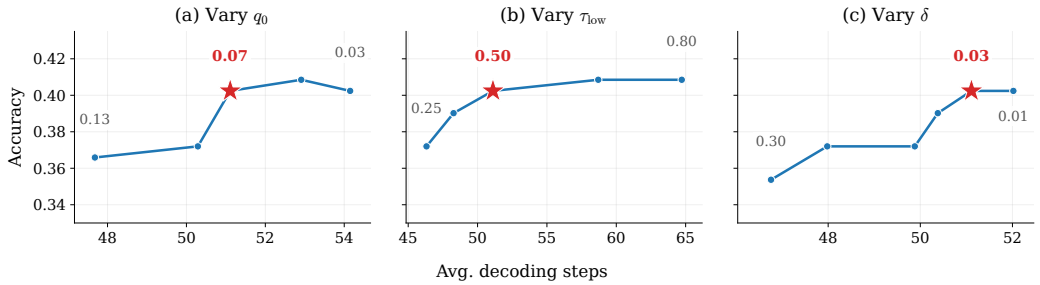


Figure 13: **One-at-a-time hyperparameter sensitivity.** LLaDA-1.5 on HumanEval. Each panel varies one of q_0, τ_{low}, or δ while fixing the other two at the reported configuration $(0.07, 0.50, 0.03)$. Points show accuracy against average decoding steps (up-left is better). The red star marks the reported configuration; labels indicate its parameter value and the endpoints of the displayed range.

J Computational Cost

Effective resistance requires a Laplacian pseudoinverse on the masked subgraph at each denoising step. Our dense symmetric eigendecomposition costs $O(|M|^3)$, compared with $O(|M|^2)$ for thresholding direct attention edges. Figure 14 reports mean per-step timings over 10 HumanEval examples on LLaDA-8B-Instruct, including graph construction, host transfer, and the eigendecomposition. At the block length $B = 32$ used throughout, $|M| \leq 32$ and resistance takes 0.73 ms of a 172 ms decoding step (0.4%). The overhead remains below 1% through $|M| = 64$, but reaches 15.6 ms (6.8%) at $|M| = 256$. For substantially larger masked sets, possible optimizations include computing resistances only for queried pairs, regularized-inverse formulations on connected graphs, or amortizing the factorization across decoding steps.

K Extended Related Work

Acceleration methods for MDMs broadly either modify the model or change only inference-time sampling. Di4C (Hayakawa et al., 2025) distills cross-position correlations. SDTT (Deschenaux & Gulcehre, 2025), Duo (Sahoo et al., 2025), FS-DFM (Karimi Monsefi et al., 2025), and dParallel (Chen et al., 2025b) train or distill for more parallel or fewer-step decoding. dUltra (Chen et al., 2025a) learns an unmasking planner. These approaches require additional model-specific training, whereas GRID leaves the base model fixed.

Among inference-time methods, the default sampler (Nie et al., 2026) and Fast-dLLM (Wu et al., 2025) use prediction confidence to choose positions to decode, EB-Sampler (Ben-Hamu et al., 2026) uses an entropy bound to determine how many positions to decode, and KLASS (Kim et al., 2026b) and LEAP (Zhang et al., 2026) use prediction stability and lookahead, respectively. DUS (Luxembourg et al., 2025) instead uses a fixed dilated schedule based on positional separation. These methods do not explicitly assess compatibility between positions decoded together.

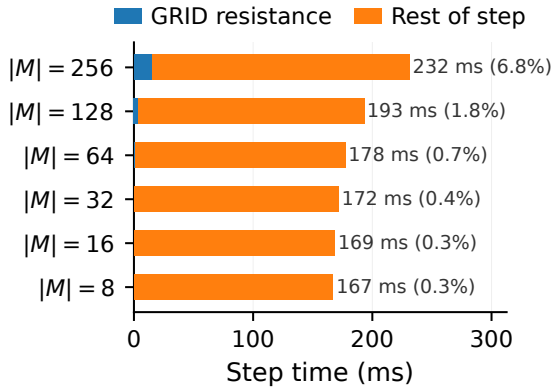


Figure 14: Per-step resistance overhead on HumanEval with LLaDA-8B-Instruct.

PUNT (Azangulov et al., 2026) temporarily reveals candidate positions and measures how the remaining masked-position marginals change, requiring $O(\log |M|)$ additional forward passes per step, while DEMASK (Ringel et al., 2026) trains a predictor from a frozen model's hidden states. Among attention-based methods, DAPD (Kim et al., 2026a) forms an attention-induced Markov random field and selects an independent set of masked positions, while DAWN (Luo et al., 2026) builds an attention dependency graph with confidence-ordered selection, anchor-guided decoding, and attention-sink filtering (Xiao et al., 2024). Both use direct attention-derived coupling between masked positions, but differ in their selection procedure: DAPD ranks candidates using a score combining prediction confidence with their node degree in the attention graph, whereas DAWN uses confidence-ordered selection. Our evaluation of DAPD's official implementation is provided in Appendix F. CLAD (Qi et al., 2026) instead groups contiguous high-confidence positions into clusters and scores attention between them. These methods do not explicitly aggregate connectivity through indirect paths in the attention-derived graph.