CARD: DENSE DENOISING FOR CAUSAL LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive language models use causal computation efficiently but generate one token at a time. Masked diffusion permits parallel generation, although through a different training interface. Applying denoising to a causal model creates an uneven-context problem: the same corruption can remove most of the useful history for one target and little for another. Causal Autoregressive Diffusion (CARD) addresses this problem while retaining the standard next-token interface. It predicts every clean token from a corrupted left prefix, using Soft Tail Masking to preserve evidence within the corrupted suffix and context-aware weights to account for the amount and location of missing history. In matched 1B-parameter experiments, CARD exceeds block diffusion (BD3LM) by 5.10 percentage points in eight-task accuracy and trains about three times as fast. Ablations attribute the gain to both mechanisms, and prediction-level analysis finds less disagreement across corrupted prefixes. The same checkpoint supports KV-cached sequential generation and confidence-based parallel decoding, with parallelism selected at inference time.

1 INTRODUCTION

Autoregressive (AR) language models combine dense next-token supervision with efficient causal attention, but standard decoding remains sequential. Masked diffusion models instead recover missing tokens in parallel (Austin et al., 2021; Sahoo et al., 2024; Nie et al., 2025), typically using bidirectional or block-structured attention that changes how context is represented and processed (Shi et al., 2024; Arriola et al., 2025a). CARD trains an ordinary causal next-token model on incomplete histories, so the resulting model can decode either sequentially or in parallel.

Corruption does not affect all causal predictions equally. Early targets have few predecessors, so a small number of masks can remove much of their available history. Later targets have longer prefixes, yet a contiguous masked suffix can still erase their nearest neighbors. A sequence-level noise rate treats these cases alike. We instead control where information is removed and how strongly each resulting prediction task contributes to training.

These choices define Causal Autoregressive Diffusion (CARD). Soft Tail Masking distributes a fixed mask budget over an expanded suffix, concentrating corruption near the end without removing every intervening token. Context-aware reweighting reduces the contribution of targets preceded by many or consecutive masks. After a one-token input shift, every clean token remains a prediction target, including tokens that were not masked. CARD preserves text order, causal attention, the next-token head, and key–value (KV) caching, and its objective does not impose a decoding block size.

We compare ten objectives at 1B parameters, with three paired seeds and 300B FineWeb tokens per run. CARD improves eight-task accuracy over uniform causal denoising by 2.93 points and over BD3LM by 5.10 points, while processing training tokens at roughly three times the BD3LM rate. Factorial and targeted controls at 1B and 110M isolate the effects of mask placement, local gaps, and target weighting. On a shared set of corruptions, CARD predicts the clean targets more accurately and varies less across incomplete prefixes.

The comparison with AR is mixed: CARD has lower clean-prefix perplexity, whereas AR retains higher downstream accuracy and stronger WinoGrande answer ranking. Domain and repeated-data experiments test whether this pattern survives outside the main training regime. We further evaluate

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.

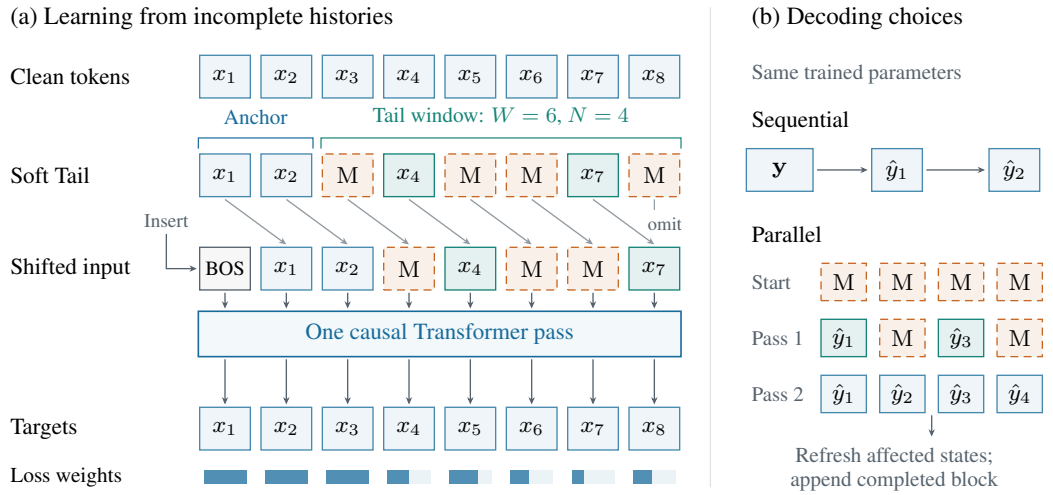


Figure 1: CARD training and decoding. The corrupted input is shifted by one token, so output n predicts clean token x_n from its strictly left-hand context. Soft Tail Masking distributes masks within a widened suffix; context-aware weights decrease as recent gaps accumulate. The same causal model supports sequential decoding and iterative commitment of multiple tokens.

six decoders on one CARD checkpoint, measuring the speed–quality trade-off of inference-time parallelism without decoder-specific retraining.

2 Dense Denoising with Causal Attention

2.1 Dense supervision from incomplete prefixes

Let $\mathbf{x} = (x_1, \ldots, x_L)$ be a clean sequence. A corruption pattern z replaces selected tokens by a reserved symbol `[MASK]`, producing $\mathbf{x}^z$. CARD shifts this sequence by one position before passing it to a causal Transformer:

$$\widetilde{\mathbf{x}}^z = (\texttt{[BOS]}, x_1^z, \ldots, x_{L-1}^z), \qquad p_\theta(x_n \mid \mathbf{x}_{<n}^z). \tag{1}$$

At output n, the newest input is x_{n-1}^z. A visible x_n therefore provides context for later targets but cannot be copied into its own prediction. All L positions receive clean-token supervision in one forward pass, with the ordinary lower-triangular attention mask.

Removing corruption recovers the usual autoregressive factorization:

$$p_\theta^{\mathrm{AR}}(\mathbf{x}) = \prod_{n=1}^{L} p_\theta(x_n \mid \mathbf{x}_{<n}). \tag{2}$$

Corrupted-prefix training, clean-prefix likelihood evaluation, and KV-cached sequential generation thus share one prediction head. During parallel decoding, unresolved predecessors occupy masked positions in the same interface.

2.2 Soft Tail Masking

Soft Tail Masking retains a clean history and visible context near each target. It separates the mask budget from the width of the eligible region. We draw $t \sim \mathcal{U}(0, 1)$ and set

$$N = \max(1, \lfloor Lt \rfloor), \qquad W = \min\{L, \max(N, \lfloor \lambda N \rfloor)\}. \tag{3}$$

We then sample exactly N positions uniformly without replacement from the final W tokens. Positions $1, \ldots, L - W$ remain clean. Within the eligible suffix, each token remains visible with probability $1 - N/W$. At high noise levels, W reaches L and the entire sequence becomes eligible for masking.

The expansion factor λ controls the balance between the clean prefix and local evidence. Setting $\lambda = 1$ masks a contiguous suffix: later targets in that suffix have no visible neighbors between themselves and its boundary. Increasing λ spreads the same mask budget across a wider region, leaving visible tokens between gaps. The default $\lambda = 1.5$ gives roughly one visible token for every two masked tokens before the window saturates. This preserves local context while concentrating corruption toward the sequence end.

This allocation gives an explicit account of the context available to each target. Let $r_n = \max\{0, n - 1 - (L - W)\}$ count eligible positions in the prefix of target n. Its expected number of visible predecessors is

$$\mathbb{E}\big[\#\text{ visible predecessors of } x_n \mid N, W\big] = \min\{n-1, L-W\} + r_n\left(1 - \frac{N}{W}\right). \tag{4}$$

The first term is the intact history before the eligible suffix; the second is the evidence retained inside it. Strict-tail masking sets the second term to zero. Soft Tail restores this local evidence without reducing the number of masks, separating corruption strength from the continuity of missing context. Appendix C derives the corresponding prefix-pattern probabilities.

2.3 Context-aware reweighting

Even with a fixed sampler, targets differ in how much of their history remains available. CARD assigns a weight to each target using the mask pattern in its prefix. Let $m_i = \mathbf{1}[x_i^z = \texttt{[MASK]}]$ and $m_0 = 0$. Define

$$S_n = \sum_{i=1}^{n-1} m_i(1 + m_{i-1})\rho^{n-1-i}, \qquad w_n = \frac{1}{\beta + S_n}. \tag{5}$$

The sum accounts for the amount of missing context, the decay ρ discounts distant masks, and $1 + m_{i-1}$ assigns extra cost to a mask that extends a consecutive gap. The recurrence $S_{n+1} = \rho S_n + m_n(1 + m_{n-1})$ computes all weights in $O(L)$ time. The weights use mask locations strictly before the target and require no additional model predictions.

For example, take an otherwise clean prefix and the defaults $\beta = 1$, $\rho = 0.5$. A mask immediately before the target gives $w_n = 1/2$; moving it three positions farther back gives $w_n = 8/9$. Two consecutive masks immediately before the target give $w_n = 2/7$. These weights distinguish local gaps from distant omissions even when a sequence-level noise rate treats them alike.

Writing $C_z(\mathbf{x}_{<n}) = \mathbf{x}^z_{<n}$, the training objective is

$$\mathcal{L}_{\text{CARD}} = \mathbb{E}_{\mathbf{x},z}\frac{1}{L}\sum_{n=1}^{L} -w_n(z)\log p_\theta(x_n \mid C_z(\mathbf{x}_{<n})). \tag{6}$$

We average over all L targets and retain the weights' absolute scale. Appendix C gives the training algorithm and the connection to independent absorbing diffusion. Section 4.2 analyzes this weighted objective directly under the fixed-budget Soft Tail distribution.

3 Experiments

3.1 Matched pretraining setup

The main study trains a 1B-parameter Transformer on FineWeb (Penedo et al., 2024), with 33 layers, hidden size 1536, 24 attention heads, context length 2048, and a 50,368-token vocabulary. Ten objectives are trained with seeds 0, 1, and 2. Within each seed, the models share aligned initial parameters, the tokenizer and ordered token stream, the AdamW schedule, and a 300B content-token budget. Each update contains about 1M content tokens. All 30 runs are evaluated at 25B, 50B, 100B, 200B, and 300B tokens.

The comparison includes clean-prefix AR, uniform causal denoising, and the CARD component controls in Table 1. MDLM and BD3LM retain their native denoising objectives and attention patterns. BD3LM uses the released FlexAttention implementation of its structured block mask (Arriola et al., 2025b; Dong et al., 2025); CARD uses standard causal attention. Timings use eight H800 GPUs, four

sequences per GPU, and 16 accumulation steps. Appendix A specifies the optimizer, baselines, and evaluation settings.

We report the macro average of accuracy on ARC-C, ARC-E, CommonsenseQA, HellaSwag, MMLU-Redux (Gema et al., 2025), PIQA, SciQ, and WinoGrande, using a shared Language Model Evaluation Harness configuration (Gao et al., 2024). Tasks are averaged within each run before averaging seeds. All contrasts pair runs by seed, and $\pm$ denotes the sample standard deviation across the three seed-level values.

Table 1: Matched 1B training suite after 300B FineWeb tokens. All 30 runs use the same parent architecture and, within seed, the same initialization and token stream. Δ is CARD minus the row method in percentage points.

Training rule	Eight-task acc. (%)	CARD − row (pp)	PPL / exp. vNLL	s / step
AR	56.20 ± 0.71	-3.07	38.8	1.90
Uniform denoising	50.20 ± 0.55	2.93	43.7	2.06
Uniform + weights	51.75 ± 0.58	1.38	39.6	2.14
Soft Tail, unit weights	51.90 ± 0.35	1.23	38.4	2.13
CARD	53.13 ± 0.28	—	34.5	2.18
Strict tail	52.15 ± 0.36	0.98	36.5	2.10
Position-only weights	52.35 ± 0.35	0.78	35.9	2.14
Position-normalized	52.63 ± 0.26	0.50	35.3	2.17
MDLM	48.35 ± 0.56	4.78	43.6†	3.28
BD3LM	48.03 ± 0.56	5.10	42.8†	6.55

Causal predictors report exact clean-prefix PPL. †MDLM and BD3LM report exponentiated Monte Carlo variational NLL. Position-only and position-normalized weights are b_n and w_n/b_n, respectively. Step time averages three 500-update post-warmup measurements on 8×H800; BD3LM uses the FlexAttention path in its released implementation.

3.2 Accuracy and training efficiency

CARD reaches 53.13% eight-task accuracy after 300B tokens, improving over uniform causal denoising by 2.93 points, MDLM by 4.78 points, and BD3LM by 5.10 points (Table 1). The CARD–BD3LM differences are positive in all three paired runs: 4.48, 5.11, and 5.72 points. AR reaches 56.20%, placing it 3.07 points above CARD.

The causal computation path also reduces training cost. CARD averages 2.18 seconds per update, compared with 2.06 for uniform causal denoising, 3.28 for MDLM, and 6.55 for BD3LM. At the common content-token update size, CARD processes training tokens approximately three times as fast as BD3LM. AR takes 1.90 seconds per update, making CARD about 15% slower than clean-prefix training.

Figure 3 shows where the accuracy differences arise. CARD gains more than ten points over BD3LM on ARC-E, PIQA, and HellaSwag; BD3LM leads by 2.36 points on MMLU-Redux. Against AR, CARD's largest deficit is HellaSwag, while MMLU-Redux is nearly unchanged. CARD therefore improves both accuracy and training throughput over the two diffusion baselines, and AR leads on both measures in this suite.

3.3 Training horizon and data reuse

CARD's advantage over BD3LM grows throughout training, from 1.84 points at 25B tokens to 5.10 points at 300B (Figure 2). The full masking-and-weighting combination takes longer to surpass its individual components. At 50B tokens, CARD trails uniform masking with context weights by 0.35 points and Soft Tail with unit weights by 0.78 points. Both comparisons reverse by 100B, after which CARD leads every non-AR variant at each checkpoint. The AR gap first widens and then narrows. Appendix A reports the complete trajectories.

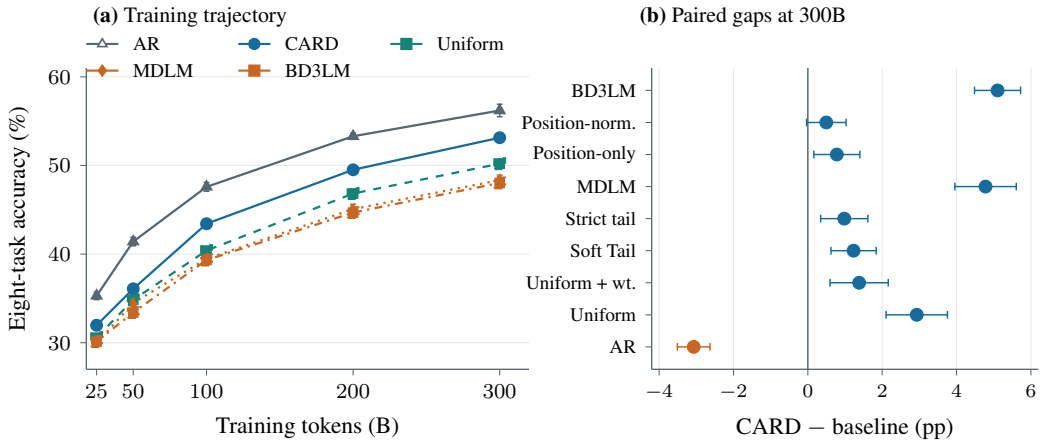


Figure 2: Matched 1B pretraining. **Left:** learning curves for representative objectives; points and whiskers show means and sample standard deviations across three paired seeds. **Right:** CARD minus each baseline at 300B tokens, calculated within seed. CARD leads every denoising baseline; AR achieves the highest downstream accuracy.

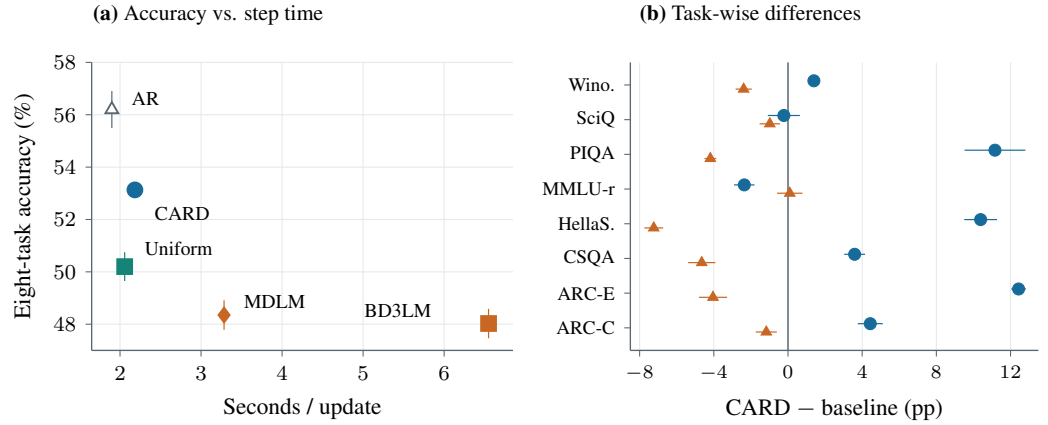


Figure 3: Accuracy and training cost at 300B tokens. **Left:** eight-task accuracy versus mean update time. **Right:** task-wise differences between CARD and AR (triangles) or BD3LM (circles), with paired-seed sample standard deviations. Positive differences favor CARD.

A separate repeated-data experiment trains 1B models for 40 passes over a fixed 1B-token FineWeb-Edu subset. CARD overtakes AR on HellaSwag at approximately pass 12; MDLM overtakes AR at pass 23 and CARD at pass 27 (Appendix D.3). The two experiments distinguish gains from consuming more training tokens from gains under repeated exposure to the same corpus. In the fixed-data study, CARD surpasses AR earlier than MDLM does, and MDLM leads later in training.

Repeated exposure differs between the two causal objectives. AR pairs each target with the same clean history on every pass. CARD resamples which predecessors remain visible, so the same text supplies different conditional prediction tasks. Soft Tail restricts this diversity to histories with an intact prefix and partially observed suffix, while context weights emphasize views with more usable nearby evidence. The exact support calculation in Appendix C characterizes these alternative histories and their probabilities under the fixed mask budget.

4 ANALYSIS

4.1 CONTRIBUTIONS OF MASKING AND WEIGHTING

A 2×2 comparison isolates Soft Tail Masking and context-aware reweighting (Table 2). At 1B, adding context weights to uniform masking improves accuracy by 1.55 points. Replacing uniform masks with Soft Tail under unit weights adds 1.70 points. Combining the two adds 2.93 points, giving an interaction of -0.32 points. Both components contribute, and their combined endpoint gain is sub-additive. The same ordering holds in a three-seed, 110M-parameter study trained on 3B LM1B tokens (Chelba et al., 2014).

Table 2: The two CARD ingredients at two scales. Rows report eight-task accuracy for the 2×2 combination of mask placement (uniform vs. Soft Tail) and token weighting (unit vs. context-dependent). Interaction is CARD − SoftTail/unit − uniform/weights + uniform/unit.

Scale	Uniform / unit	Uniform / weights	Soft Tail / unit	CARD	Interaction
110M / LM1B / 3B	43.87	45.00	45.22	46.47	0.12
1B / FineWeb / 300B	50.20	51.75	51.90	53.13	-0.32

Table 3: Targeted component contrasts (accuracy points). Each difference is formed within seed; entries are mean ± sample SD. The consecutive-gap ablation changes $m_i(1 + m_{i-1})$ to m_i at 110M.

Contrast	110M / LM1B / 3B	1B / FineWeb / 300B
CARD − strict tail	0.98 ± 0.27	0.98 ± 0.64
Position-normalized − unit	0.69 ± 0.26	0.73 ± 0.13
CARD − position-only	0.77 ± 0.17	0.78 ± 0.62
CARD − position-normalized	0.57 ± 0.17	0.50 ± 0.53
CARD − no consecutive-gap factor	0.59 ± 0.05	—

Targeted controls identify the value of local context (Table 3). Strict-tail masking retains the mask budget and weight formula but sets $\lambda = 1$, removing visible tokens within the suffix. This costs 0.98 accuracy points at both scales. At 110M, removing the consecutive-gap factor $1 + m_{i-1}$ costs 0.59 points. That ablation increases both the mean weight, from 0.612 to 0.661, and the gradient norm, from 0.83 to 0.86.

We separate mask-dependent weighting from average positional scaling using a frozen profile $b_n \approx \mathbb{E}_z w_n(z)$, estimated from 50,000 calibration draws. Position-only weights use b_n; position-normalized weights use w_n/b_n. Both retain Soft Tail masks. On a disjoint 10,000-draw bank, the largest position-wise normalization error is 0.35%. Position-normalized weights improve over unit weights by 0.73 points at 1B and 0.69 points at 110M, so mask-dependent weighting retains a benefit after removing its average positional scale. Position-only weights fall short of full CARD by 0.78 and 0.77 points, respectively. Appendix B.4 derives this separation.

The 110M parameter sweep gives 46.03–46.24% accuracy for nearby settings of λ, ρ, and β, compared with 46.47% for the default. Appendix A reports the settings and paired differences.

4.2 PREDICTION CONSISTENCY ACROSS CORRUPTED PREFIXES

To connect the component gains to model predictions, we decompose the weighted loss into target fit and disagreement across corruptions. Fix a clean sequence and target n. Let Q be the corruption law, $a_n = \mathbb{E}_Q w_n$, and $Q_n^w(z) = Q(z)w_n(z)/a_n$. Writing $p_{n,z}(v) = p_\theta(v \mid C_z(\mathbf{x}_{<n}))$, define the normalized geometric consensus

$$\bar{p}_n(v) \propto \exp\{\mathbb{E}_{Q_n^w} \log p_{n,z}(v)\}. \tag{7}$$

The expected weighted loss then satisfies

$$\mathbb{E}_Q[-w_n \log p_{n,z}(x_n)] = a_n\,[-\log \bar{p}_n(x_n) + J_n]\,, \quad J_n = \mathbb{E}_{Q_n^w}\, \mathrm{D}_{\mathrm{KL}}(\bar{p}_n \| p_{n,z}). \tag{8}$$

The first term measures how well the consensus predicts the clean token. The nonnegative term J_n measures disagreement among predictions made from different corrupted prefixes. This identity holds for the fixed-budget Soft Tail law; Appendix B gives the proof and logit-space form.

Equation (8) also separates the two effects of context-aware weighting. The factor a_n sets a target's average contribution, whereas Q_n^w determines which corrupted prefixes receive greater influence. A position-only weight changes the first factor but leaves the view distribution at Q. Dividing w_n by a fixed positive positional profile preserves Q_n^w while changing the average scale. These are precisely the interventions tested by the controls in Section 4.1. Because w_n decreases with the missing-context score S_n, the weighted view law emphasizes prefixes with more usable recent history. Appendix B proves that $\mathbb{E}_{Q_n^w} S_n \leq \mathbb{E}_Q S_n$.

We evaluate the 110M checkpoints on a shared bank of 2,048 targets, each with 32 corrupted prefixes. Uniform causal denoising has mean consensus loss $F_w = 3.498$ nats and disagreement $J_w = 0.183$. Soft Tail with unit weights reduces these values to 3.347 and 0.154; full CARD reduces them to 3.205 and 0.111. Individual-view top-1 accuracy rises from 34.3% to 38.0%. The ordering persists with 64 views. Because every model receives the same evaluation corruptions, these differences reflect their learned predictions: CARD fits the target better and varies less across missing-context patterns.

4.3 Clean-prefix likelihood and answer ranking

CARD's improvements in denoising also extend to clean-prefix likelihood. At 300B tokens, its PPL is 34.5, compared with AR's 38.8, an 11.1% reduction (Table 4). Every paired seed has lower PPL and lower downstream accuracy at both 100B and 300B. Likelihood evaluation uses the same tokenizer and 12.8M held-out tokens, exact teacher forcing, disabled dropout, and unit weights. The 0.117-nat token-NLL improvement therefore measures prediction on clean text.

Table 4: Exact clean-prefix PPL (↓) and eight-task accuracy (↑). Mean ± sample SD over three paired seeds. The 1B runs favor CARD on PPL and AR on accuracy in every pair; 110M accuracy has mixed seed-level signs.

Setting	Model	PPL	Accuracy (%)
110M / LM1B / 3B	AR	22.35 ± 0.11	46.59 ± 0.36
	CARD	21.98 ± 0.10	46.47 ± 0.28
1B / FineWeb / 100B	AR	51.20 ± 0.08	47.58 ± 0.50
	CARD	47.60 ± 0.36	43.43 ± 0.55
1B / FineWeb / 300B	AR	38.80 ± 0.54	56.20 ± 0.71
	CARD	34.50 ± 0.37	53.13 ± 0.28

The independent 1B model-family study shows the domain distribution of this gain. CARD improves clean-prefix PPL over AR on six of eight domains, including LM1B, LAMBADA, and PTB; AR leads on arXiv and PubMed (Appendix D). At 110M in the 3B-token LM1B component study, CARD also has lower PPL in all three seeds, while its mean downstream difference from AR is -0.12 points with mixed seed-level signs. The paired seed results are in Table 15.

We examine the 1B likelihood–accuracy difference through WinoGrande candidate scores (Sakaguchi et al., 2020). For the correct and incorrect candidates, define $m = s_{\text{correct}} - s_{\text{wrong}}$ and the two-choice log loss $\log(1 + e^{-m})$. The margin measures preference for the correct answer; the log loss evaluates its probability after normalizing over the two candidates. Both are computed per question before averaging.

Table 5: WinoGrande answer ranking on 1,267 validation questions. The signed margin favors the correct candidate; two-choice log loss normalizes the candidate scores. Entries are mean ± sample SD over three paired seeds.

Tokens (B)	Model	Accuracy (%)	Margin ↑	Choice log loss ↓
100	AR	53.43 ± 0.34	0.038 ± 0.004	0.712 ± 0.003
100	CARD	52.46 ± 0.32	0.027 ± 0.010	0.717 ± 0.003
300	AR	55.67 ± 0.72	0.053 ± 0.012	0.704 ± 0.006
300	CARD	53.28 ± 0.42	0.023 ± 0.008	0.718 ± 0.004

At 300B, CARD's mean signed margin is 0.023 versus AR's 0.053, and its choice log loss is 0.718 versus 0.704 (Table 5). The mean absolute margins are nearly equal, 0.485 and 0.488. The models separate candidates by similar amounts but differ in which candidate they favor. Every seed shows smaller signed margins and higher choice loss for CARD at both budgets. At 300B, AR's net advantage is 24–34 of 1,267 questions per seed; Appendix A.4 reports the paired decision changes. Lower corpus PPL thus coexists with weaker answer discrimination under both accuracy and a smooth task-level loss.

5 Parallel Decoding from a Single Checkpoint

CARD's corrupted-prefix conditionals predict future tokens before every predecessor is known. We use them for confidence-based parallel decoding without further training. A candidate block begins with K masks. Each pass commits greedy predictions whose probabilities exceed τ and recomputes the unresolved positions using the updated block. After $T_{\max}$ passes, the remaining positions receive greedy predictions. The completed block is added to the causal KV cache after a refresh. Block size is chosen at inference and is independent of the training mask budget.

We compare sequential greedy decoding, Jacobi ($K = 8$), verified Lookahead (Fu et al., 2024), and CARD with 8-, 16-, or 32-token blocks. All six use the same 1B checkpoint, prompts, tokenizer, and stopping rules. Timing includes prefill, verification, cache refresh, and finalization; Appendix E gives the complete protocol.

Table 6: Six decoders applied to the same 1B CARD checkpoint. Throughput is measured at batch 128 over 20 timing windows; latency is median batch-1 completion time. CARD-K denotes confidence decoding with $(K, T_{\max}) = (8, 8), (16, 16), (32, 8)$ and $\tau = 0.9$.

Decoder	Tokens/s	Speedup	Latency (s)	QA F1
Sequential	3906 ± 46	1.00×	1.84	21.94
Jacobi	3756 ± 35	0.96×	1.94	21.94
Lookahead	4768 ± 75	1.22×	1.55	21.94
CARD-8	5515 ± 79	1.41×	1.30	21.64
CARD-16	6640 ± 88	1.70×	1.15	21.27
CARD-32	14975 ± 277	3.83×	0.89	20.24

Table 7: Eight-token CARD relative to sequential CARD. Continuation quality reports the range of external-score changes across Qwen3-8B, SmolLM3-3B, Gemma-3-27B, and GPT-2 Large (positive is worse); TriviaQA reports F1 change. Agreement is exact output-token agreement, and edit distance is normalized by the longer output.

Dataset	Quality change	Agreement (%)	Token edit
HellaSwag	E_s: +2.58 to +2.89%	88.3	0.027
WikiText-103	E_s: +2.96 to +3.41%	84.7	0.033
PG-19	E_s: +3.38 to +3.42%	81.2	0.034
TriviaQA	F1: −0.30 pp	87.1	0.042

Jacobi and Lookahead reproduce all 4,500 sequential outputs exactly; Lookahead increases batch-128 throughput by $1.22\times$. CARD-8 reaches $1.41\times$, with TriviaQA F1 changing from 21.94 to 21.64. Across the three continuation datasets, 81–88% of CARD-8 outputs match sequential decoding exactly, and normalized token edit distance is 0.027–0.034. External continuation scores from Qwen3-8B, SmolLM3-3B, Gemma-3-27B, and GPT-2 Large increase by 2.6–3.4%, indicating lower continuation likelihood.

Increasing block size yields more throughput and larger output changes. CARD-32 reaches $3.83\times$ throughput; TriviaQA F1 falls to 20.24, and HellaSwag continuation scores increase by 38–39%. The checkpoint therefore supports distinct operating points: verified Lookahead preserves the sequential output, CARD-8 trades 0.30 F1 points for higher throughput, and CARD-32 prioritizes speed with a larger quality loss.

6 Related Work

Denoising objectives. Discrete diffusion models learn reverse transitions from corrupted categorical states (Austin et al., 2021; Lou et al., 2024). MDLM and MD4 simplify masked training objectives, and LLaDA scales masked diffusion to larger language models (Sahoo et al., 2024; Shi et al., 2024; Nie et al., 2025). BD3LM combines autoregression across blocks with denoising inside each block (Arriola et al., 2025a). CARD instead denoises strictly left-hand context and supervises every clean target through a shifted next-token interface.

Causal diffusion interfaces. FLUID and WeDLM also retain causal attention, but their conditioning rules and training objectives differ from CARD's (Table 8). FLUID adapts an AR model using AR and masked-span losses, then trains a horizon-selection head on a frozen backbone (Ma et al., 2026). WeDLM places observed tokens before masks while retaining logical position IDs; a masked token can therefore access observed tokens to its logical right within a block (Liu et al., 2025). CARD preserves logical text order and studies dense denoising, mask placement, and target weighting from a shared initialization.

Table 8: Causal denoising interfaces. FLUID (Ma et al., 2026) and WeDLM (Liu et al., 2025) adapt pretrained AR models; CARD is evaluated in matched pretraining runs. The methods differ in token visibility, supervision, and decoder adaptation.

	CARD	FLUID	WeDLM
Initialization	Shared 1B parent tensors in matched pretraining	Pretrained openPangu-Embedded-7B AR backbone	Pretrained Qwen2.5-7B / Qwen3-8B AR backbones
Conditioning	Corrupted left prefix in original token order	Strictly causal history with masked spans	Observed tokens reordered before masks; within-block logical right context is visible
Targets	Every clean next token after a one-token shift	Observed-token AR prediction and masked-span reconstruction	Masked-token reconstruction and auxiliary AR targets
Objective	Dense cross-entropy with Soft Tail masks and context weights	Hybrid AR/denoising loss; separate horizon-head calibration	Masked loss weighted by $1/\gamma$ plus AR loss; dual-stream training
Decoding	Cached next-token prediction; optional confidence blocks	Learned K-Head selects denoising horizon	Reordered streaming window with prefix commitment

Parallel decoding. ReFusion and D2F combine diffusion updates with AR structure (Li et al., 2026; Wang et al., 2026). Consistency LMs and Jacobi Forcing train models for parallel decoding trajectories (Kou et al., 2024; Hu et al., 2026), while Lookahead proposes and verifies tokens from a fixed causal model (Fu et al., 2024). ParallelBench evaluates the resulting speed–quality trade-offs (Kang et al., 2026). Our decoder comparison measures parallel generation from pretrained CARD conditionals before any decoder-specific training.

7 Conclusion

CARD trains a standard causal language model to denoise incomplete left contexts. Paired experiments and controls at two model scales show that both Soft Tail Masking and context-aware reweighting contribute to the improvement. CARD is more accurate and faster to train than MDLM and BD3LM in our matched study. Relative to AR, it achieves lower clean-prefix PPL but lower downstream accuracy. Its corrupted-prefix conditionals also enable cached sequential generation and adjustable parallel decoding from the same checkpoint, without architectural changes or decoder-specific retraining.

Ethics statement

CARD inherits the risks of large-scale language-model pretraining and generation, including biased or private training content and inaccurate or harmful outputs. The experiments evaluate accuracy, likelihood, and throughput; factuality, privacy, and safety require separate evaluation.

Reproducibility statement

Appendix A specifies training and evaluation, Appendices B and C provide proofs, and Appendix E defines the decoders. The supplementary package includes run-level results, paired likelihood and answer-ranking analyses, recorded generation outputs, and scripts for regenerating the numerical tables and figures. It also records the provenance of the independent domain results and repeated-data summary.

AI use statement

Generative AI tools were used for language editing. The authors reviewed the resulting text and take responsibility for all claims, analyses, and artifacts.

References

Marianne Arriola, Aaron Kerem Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=tyEyYT267x.

Marianne Arriola, Aaron Kerem Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. BD3-LMs: Official implementation of block diffusion language models. GitHub repository, 2025b. URL https://github.com/kuleshov-group/bd3lms. Training examples use the FlexAttention backend for BD3-LM.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Elie Bakouch, Carlos Miguel Patiño, Anton Lozhkov, Edward Beeching, Aymeric Roucher, Nouamane Tazi, et al. SmolLM3: smol, multilingual, long-context reasoner. Hugging Face, 2025. URL https://huggingface.co/blog/smollm3.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Philipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. In *Interspeech 2014*, pp. 2635–2639, 2014. doi: 10.21437/Interspeech.2014-564. URL https://www.isca-archive.org/interspeech_2014/chelba14_interspeech.html.

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flexattention: A programming model for generating fused attention variants. In *Proceedings of Machine Learning and Systems*, volume 7, 2025. URL https://proceedings.mlsys.org/paper_files/paper/2025/hash/61a9278dfef5f871b5e472389f8d6fa1-Abstract-Conference.html.

Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. Break the sequential dependency of LLM inference using lookahead decoding. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 14060–14079, 2024.

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

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile Van Krieken, and Pasquale Minervini. Are we done with MMLU? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5069–5096, Albuquerque, New Mexico, apr 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.262. URL https://aclanthology.org/2025.naacl-long.262/.

Gemma Team. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Sebastian Gruber and Florian Buettner. Uncertainty estimates of predictions via a general bias-variance decomposition. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 11331–11354, 2023.

Lanxiang Hu, Siqi Kou, Yichao Fu, Samyam Rajbhandari, Tajana Rosing, Yuxiong He, Zhijie Deng, and Hao Zhang. Fast and accurate causal parallel decoding using jacobi forcing. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2512.14681. arXiv:2512.14681.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pp. 1601–1611, 2017. doi: 10.18653/v1/P17-1147.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjae Lee, Yuchen Zeng, Shuibai Zhang, Coleman Hooper, Yuezhou Hu, Hyung Koo, Nam Ik Cho, and Kangwook Lee. ParallelBench: Understanding the trade-offs of parallel decoding in diffusion LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/9156b0f6dfa9bbd18c79cc459ef5d61c-Abstract-Conference.html.

Siqi Kou, Lanxiang Hu, Zhezhi He, Zhijie Deng, and Hao Zhang. CLLMs: Consistency large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 25426–25440, 2024.

Jia-Nan Li, Jian Guan, Wei Wu, and Chongxuan Li. ReFusion: A diffusion large language model with parallel autoregressive decoding. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/585979c057a1b30796cf317063559638-Abstract-Conference.html.

Aiwei Liu, Minghua He, Shaoxun Zeng, Sijun Zhang, Linhao Zhang, Chuhan Wu, Wei Jia, Yuan Liu, Xiao Zhou, and Jie Zhou. WeDLM: Reconciling diffusion language models with standard causal attention for fast inference. *arXiv preprint arXiv:2512.22737*, 2025.

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

Xiangyu Ma, Teng Xiao, Zuchao Li, and Lefei Zhang. From AR to diffusion: Efficiently adapting large language models with strictly causal and elastic horizons. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20914–20927, San Diego, California, United States, jul 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.958. URL https://aclanthology.org/2026.acl-long.958/.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 56354–56392, 2025. doi: 10.52202/085713-1689. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/48b383b24230e0e6e649d9c98dae4d8c-Abstract-Conference.html.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sMyXP8Tanm.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849. Curran Associates, Inc., 2024. doi: 10.52202/079017-0970.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylKikSYDH.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184. Curran Associates, Inc., 2024. doi: 10.52202/079017-4135.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6399.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 103131–103167. Curran Associates, Inc., 2024. doi: 10.52202/079017-3277.

Xu Wang, Chenkai Xu, Yijie Jin, Jiachun Jin, Hao Zhang, Kai Yu, and Zhijie Deng. Diffusion LLMs can do faster-than-AR inference via discrete diffusion forcing. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/04bb76a98d9f32226b3beba7bd26a51f-Abstract-Conference.html.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019. doi: 10.18653/v1/P19-1472.

A Training and Evaluation Settings

A.1 Matched 1B pretraining

The FineWeb study (Penedo et al., 2024) uses a 1B-parameter parent architecture with 33 Transformer layers, hidden size 1536, intermediate size 4096, 24 attention heads, a vocabulary of 50,368 tokens, and context length 2048. Within each of seeds 0, 1, and 2, all ten variants start from aligned copies of the same parameters and read the same tokenized documents in the same order. Tokenization, optimization, training budget, checkpoints, and evaluation are shared. MDLM uses bidirectional denoising attention, BD3LM uses block diffusion, and the remaining objectives use the shifted causal interface.

The causal controls are defined as follows. AR uses ordinary clean-prefix teacher forcing. Uniform causal denoising uses the shifted interface in Eq. (1), masks uniformly, and assigns unit target weights. "Uniform + weights" keeps those mask locations but applies w_n from Eq. (5). "Soft Tail, unit weights" uses Eq. (3) with $\lambda = 1.5$ but assigns every target weight one. CARD combines Soft Tail with $(\beta, \rho) = (1, 0.5)$. Strict tail sets $\lambda = 1$. The position-only and position-normalized controls use the same Soft Tail views with the weights described in Appendix B.4. MDLM and BD3LM retain their method-native objectives. BD3LM uses block size 32 and a frozen 256-bin monotone noise schedule; the schedule file is included with the supplementary records.

BD3LM uses `model.attn_backend=flex`, the custom block-mask path in the released implementation (Arriola et al., 2025b; Dong et al., 2025). Its training times include this structured attention computation.

All 30 runs train to 300B non-padding content tokens and are evaluated at 25B, 50B, 100B, 200B, and 300B. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.1, and gradient clipping at 1.0. The learning rate warms up for 2,500 updates to 3×10^{-4} and then follows cosine decay to 3×10^{-5}. Training uses bfloat16 on eight H800 80GB GPUs, four sequences per device, and 16 accumulation steps; one full update contains 1,048,576 content tokens. Step time is measured over 500 post-warmup updates for every run. The synchronized timing table in the main text averages the three run-level means for each variant.

Table 9: Task-level means (%) for the controlled 1B matrix at 300B tokens. Each entry averages three paired seeds; the last column is the unweighted mean over the eight tasks.

Training rule	ARC-C	ARC-E	CSQA	HellaS.	MMLU-r	PIQA	SciQ	Wino.	Mean
AR	33.61	64.60	52.97	60.84	25.22	75.85	80.84	55.67	56.20
Uniform denoising	30.54	55.83	43.93	49.03	25.13	66.91	78.09	52.15	50.20
Uniform + weights	31.31	57.82	46.46	51.12	25.52	69.47	79.16	53.14	51.75
Soft Tail, unit weights	31.79	58.13	46.62	51.21	25.39	69.45	79.41	53.20	51.90
CARD	32.44	60.55	48.32	53.60	25.32	71.66	79.86	53.28	53.13
Strict tail	31.80	58.48	47.41	51.77	25.33	69.42	79.51	53.48	52.15
Position-only weights	32.07	59.03	47.38	52.21	25.32	70.02	79.62	53.14	52.35
Position-normalized	32.19	59.98	47.14	52.70	25.43	70.76	79.86	53.01	52.63
MDLM	30.01	50.05	37.11	49.46	27.16	60.62	77.19	55.19	48.35
BD3LM	28.01	48.12	44.73	43.22	27.68	60.50	80.08	51.88	48.03

Table 10: Eight-task accuracy across the controlled 1B training trajectory. Entries are means over three paired seeds.

Training rule	25B	50B	100B	200B	300B
AR	35.29	41.41	47.58	53.29	56.20
Uniform denoising	30.57	34.84	40.40	46.83	50.20
Uniform + weights	31.31	36.43	42.00	48.23	51.75
Soft Tail, unit weights	31.78	36.86	42.38	48.44	51.90
CARD	31.97	36.08	43.43	49.51	53.13
Strict tail	32.13	37.15	42.60	48.76	52.15
Position-only weights	31.60	36.80	42.62	49.01	52.35
Position-normalized	31.97	36.94	42.80	49.01	52.63
MDLM	29.96	34.49	39.47	45.07	48.35
BD3LM	30.12	33.40	39.33	44.68	48.03

Table 11: How the endpoint comparisons develop with training. The first row is CARD accuracy; the remaining rows are CARD minus the named baseline in percentage points. Negative values favor the baseline.

Comparison	25B	50B	100B	200B	300B
CARD accuracy	31.97	36.08	43.43	49.51	53.13
vs. uniform	1.39	1.24	3.03	2.68	2.93
vs. uniform + weights	0.65	-0.35	1.43	1.28	1.38
vs. Soft Tail / unit	0.19	-0.78	1.04	1.07	1.23
vs. BD3LM	1.84	2.68	4.10	4.83	5.10
vs. AR	-3.32	-5.33	-4.16	-3.78	-3.07

Table 12 reports all six seed-level endpoint task vectors for CARD and BD3LM.

Table 12: Complete task accuracy (%) for the six 1B/300B-token runs. Seed identifiers pair the corresponding initial parameters and data stream.

Model	Seed	ARC-C	ARC-E	CSQA	HellaS.	MMLU-r	PIQA	SciQ	Wino.
CARD	0	32.13	60.40	48.31	53.75	25.15	70.51	79.59	53.12
CARD	1	32.93	61.01	48.33	53.67	25.35	72.32	80.00	53.75
CARD	2	32.27	60.25	48.33	53.39	25.46	72.15	79.99	52.96
BD3LM	0	28.45	47.95	44.93	44.02	28.04	61.24	80.40	52.09
BD3LM	1	27.95	48.98	45.16	43.63	27.74	60.14	80.63	52.25
BD3LM	2	27.62	47.43	44.10	42.00	27.25	60.12	79.22	51.30

A.2 110M LM1B CONTROLS

The smaller component study trains 110M-parameter models on the One Billion Word Benchmark (LM1B) (Chelba et al., 2014). The architecture has ten layers, hidden size 768, intermediate size 3072, twelve attention heads, and the same vocabulary and context length as the 1B model. Each configuration consumes 3B non-padding content tokens. Causal variants use the shifted prediction interface; MDLM predicts masked tokens with bidirectional attention. The position-only and position-normalized controls share CARD's Soft Tail sampler.

AdamW uses the same momentum, numerical epsilon, weight decay and gradient clipping as above. The learning rate has 600 warmup updates, peak 5×10^{-4} and cosine decay to 5×10^{-6}. Eight workers use four sequences per microbatch and four accumulation steps. The full-update capacity is 262,144 token positions; stopping is determined by the content-token budget. Within each seed, the

initialization and token order are paired across configurations, with separate random streams for data, masks, and dropout.

Both controlled studies use `CARD-BPE-50368-v1`, a byte-pair encoding tokenizer with NFC normalization. BOS, EOS, mask, and padding have reserved IDs 0–3. Tables identify models by training rule; the accompanying manifests record run identifiers, tokenization fingerprints, and checkpoint references.

A.3 Metrics and paired aggregation

The eight-task score weights ARC-C, ARC-E, CommonsenseQA, HellaSwag, MMLU-Redux (Gema et al., 2025), PIQA, SciQ and WinoGrande equally. Evaluation follows the Language Model Evaluation Harness (Gao et al., 2024); task versions, prompts, shots, and scoring procedures are shared across each controlled comparison. Per-seed scores are averaged only after calculating the eight-task mean. For the 110M WinoGrande runs we use the `winogrande_xl` validation split with five-shot prompts, replace the sentence blank by each candidate, map dataset labels `1`/`2` to candidate indices 0/1, and choose the candidate with higher continuation log likelihood. The same label convention is used for all reported WinoGrande scores.

For each paired contrast, $d_s = u_s - v_s$ is computed within seed. Its sample SD is

$$s_d = \sqrt{\frac{1}{3-1}\sum_{s=0}^{2}(d_s - \bar{d})^2}. \tag{9}$$

Table 13 shows the three individual differences. At 300B tokens in the 1B comparison they are 4.48, 5.11, and 5.72 points. The corresponding 5.10 ± 0.62 summary describes variation across these paired runs. Paired covariance enters through $s_d^2 = s_u^2 + s_v^2 - 2s_{uv}$.

Clean-prefix PPL is $\exp(\mathrm{NLL}_{\mathrm{total}}/T)$ under teacher forcing, with dropout disabled and unit evaluation weights. Each controlled causal evaluation uses $T = 12.8$M valid tokens. We reconstruct PPL from each run's total NLL before averaging across seeds. The diffusion models instead report exponentiated Monte Carlo variational NLL, labeled separately in each table.

Table 13: Paired accuracy differences (percentage points). Each seed difference is formed before computing its mean and sample SD. The first row is the controlled 1B/300B study; the remaining rows are the 110M/3B study.

Comparison	Seed 0	Seed 1	Seed 2	Mean ± SD
1B/300B: CARD vs. BD3LM	4.48	5.11	5.72	5.10 ± 0.62
CARD vs. uniform causal	2.18	2.62	3.03	2.61 ± 0.43
CARD vs. MDLM	3.48	4.08	4.39	3.98 ± 0.46
CARD vs. AR	0.43	-0.82	0.04	-0.12 ± 0.64
Soft Tail vs. strict tail	0.70	1.01	1.22	0.98 ± 0.27
Soft Tail vs. uniform masks	1.12	1.52	1.78	1.47 ± 0.33
Context vs. unit weights	0.89	1.20	1.68	1.26 ± 0.40
Mask–weight interaction	-0.17	0.09	0.43	0.12 ± 0.30
Position-only vs. unit	0.16	0.57	0.72	0.48 ± 0.29
Position-normalized vs. unit	0.40	0.75	0.92	0.69 ± 0.26
CARD vs. position-only	0.73	0.63	0.96	0.77 ± 0.17
CARD vs. position-normalized	0.49	0.45	0.76	0.57 ± 0.17
Position–weight interaction	0.33	-0.12	0.04	0.08 ± 0.23

Table 14: Task-level accuracy means (%) in the 110M/3B-token study. Every row averages three runs; seed-level scores are provided in the supplementary data.

Configuration	ARC-C	ARC-E	CSQA	HellaS.	MMLU-r	PIQA	SciQ	Wino.	Mean
AR	28.24	52.25	40.71	44.62	25.55	62.52	65.61	53.22	46.59
Uniform causal	24.77	49.64	37.75	42.51	23.59	59.28	63.07	50.30	43.87
Uniform + weights	26.34	50.92	39.28	43.30	24.42	60.76	63.67	51.33	45.00
Soft Tail (unit weights)	26.88	51.02	39.42	43.50	24.48	60.98	63.83	51.64	45.22
CARD	28.15	52.31	40.80	45.02	25.03	62.28	65.29	52.91	46.47
Strict tail	27.20	50.88	39.73	43.66	24.67	61.55	64.54	51.75	45.50
MDLM	23.85	47.80	36.36	40.02	22.39	58.10	61.19	50.22	42.49
Position-only weights	26.74	51.53	40.24	44.06	24.76	61.63	64.53	52.12	45.70
Position-normalized	27.30	51.62	40.08	44.59	24.69	61.71	64.73	52.54	45.91

A.4 Likelihood and answer-ranking diagnostics

Table 15 reports the exact clean-prefix PPL and eight-task score for each paired AR/CARD run. PPL is reconstructed from total NLL and the 12.8M evaluation-token count before aggregation. The 110M rows refer to the 3B-token, context-2048 LM1B component study; the separate 33B-token, context-128 EMA study is in Table 25.

Table 15: Every paired AR/CARD likelihood and accuracy result. PPL is reconstructed from total NLL and the evaluation-token count. Δ is CARD minus AR accuracy in percentage points.

Seed	AR PPL	CARD PPL	AR acc. (%)	CARD acc. (%)	Δ (pp)
110M / LM1B / 3B					
0	22.336	21.977	46.30	46.73	0.43
1	22.247	21.883	46.99	46.17	-0.82
2	22.467	22.080	46.48	46.52	0.05
1B / FineWeb / 100B					
0	51.238	48.012	47.10	43.04	-4.06
1	51.103	47.339	48.10	44.06	-4.04
2	51.259	47.449	47.55	43.18	-4.37
1B / FineWeb / 300B					
0	38.413	34.926	55.45	52.87	-2.58
1	38.566	34.247	56.85	53.42	-3.43
2	39.421	34.327	56.30	53.10	-3.20

The WinoGrande diagnostic uses all 1,267 validation questions at each seed and budget. The per-question data contain the predicted option, gold option, and nonnegative winning-versus-losing log-likelihood gap d_i. We recover the correct-answer margin as $m_i = (2c_i - 1)d_i$, where c_i is one for a correct prediction and zero otherwise. This is the difference between the correct and incorrect candidate scores. We compute two-choice log loss with the numerically stable expression $\mathrm{softplus}(-m_i)$ and average it over questions, rather than applying softplus after averaging margins. The absolute margin $|m_i|$ measures the separation between the two candidates irrespective of correctness.

Table 16: Seed-level WinoGrande diagnostic. All entries average the same 1,267 questions. m is the correct-minus-incorrect candidate score; $|m|$ measures score separation irrespective of correctness. Choice log loss is averaged over questions.

Tokens (B)	Seed	Model	Acc. (%)	Mean m	Mean $\|m\|$	Choice log loss
100	0	AR	53.20	0.0339	0.4886	0.7141
100	0	CARD	52.17	0.0167	0.4843	0.7208
100	1	AR	53.83	0.0374	0.4970	0.7128
100	1	CARD	52.80	0.0300	0.4833	0.7150
100	2	AR	53.28	0.0426	0.4843	0.7083
100	2	CARD	52.41	0.0351	0.5030	0.7148
300	0	AR	55.01	0.0396	0.4868	0.7103
300	0	CARD	53.12	0.0161	0.4820	0.7217
300	1	AR	56.43	0.0565	0.4870	0.7023
300	1	CARD	53.75	0.0323	0.4877	0.7141
300	2	AR	55.56	0.0623	0.4908	0.6994
300	2	CARD	52.96	0.0215	0.4844	0.7187

At 300B, AR answers 266, 268, and 276 questions correctly that CARD misses in seeds 0, 1, and 2; CARD answers 242, 234, and 243 correctly that AR misses. The net differences are 24, 34, and 33 of 1,267 questions. At 100B, the corresponding net differences are 13, 13, and 11 questions. Every pair therefore favors AR in accuracy, mean correct-answer margin, and two-choice log loss. Table 16 reports these task-level comparisons alongside the corpus-likelihood results in Table 15.

A.5 Parameter sweeps

Each sweep changes one component of the default $(\lambda, \rho, \beta) = (1.5, 0.5, 1)$ while retaining three seeds and the 3B-token budget. Removing the consecutive-gap factor replaces $m_i(1 + m_{i-1})$ by m_i. The default and strict-tail rows reuse the corresponding runs from the component study. Table 17 reports accuracy, paired differences, average weight, and gradient norm; seed-level records are included in the supplementary data.

Table 17: One-factor changes at 110M/3B tokens. Unchanged parameters retain $\lambda = 1.5$, $\rho = 0.5$, $\beta = 1$ and the consecutive-mask term. Mean weight and gradient norm describe the implemented training rule.

Setting	Accuracy (%)	Δ (pp)	Mean weight	Grad. norm
$\lambda = 1$ (strict tail)	45.50 ± 0.47	-0.98 ± 0.27	0.55	0.79
$\lambda = 1.2$	46.24 ± 0.20	-0.24 ± 0.09	0.59	0.81
Default CARD	46.47 ± 0.28	0.00 ± 0.00	0.61	0.83
$\lambda = 2$	46.24 ± 0.23	-0.24 ± 0.06	0.63	0.84
$\rho = 0.25$	46.20 ± 0.24	-0.28 ± 0.04	0.68	0.85
$\rho = 0.75$	46.10 ± 0.26	-0.38 ± 0.05	0.52	0.78
No consecutive-gap factor	45.88 ± 0.24	-0.59 ± 0.05	0.66	0.86
$\beta = 0.5$	46.13 ± 0.25	-0.34 ± 0.04	0.89	0.91
$\beta = 2$	46.03 ± 0.28	-0.45 ± 0.02	0.37	0.71

B Analysis of Weighted Corrupted-prefix Prediction

B.1 Proof of the fitting–consistency identity

Fix a clean sequence and target. Write $p_z(v) = p_\theta(v \mid C_z(\mathbf{x}_{<n}))$, $a = \mathbb{E}_Q w > 0$, and $q(z) = Q(z)w(z)/a$. Finite logits give positive probabilities on the finite vocabulary. Define

$$g(v) = \mathbb{E}_q \log p_z(v), \qquad Z = \sum_v e^{g(v)}, \qquad \bar{p}(v) = \frac{e^{g(v)}}{Z}. \tag{10}$$

Then

$$\mathbb{E}_Q[-w \log p_z(x_n)] = -ag(x_n) = a[-\log \bar{p}(x_n) - \log Z]. \tag{11}$$

Moreover,

$$\mathbb{E}_q \, \mathrm{D}_{\mathrm{KL}}(\bar{p} \| p_z) = \sum_v \bar{p}(v) \left[\log \bar{p}(v) - \mathbb{E}_q \log p_z(v)\right] \tag{12}$$

$$= -\log Z. \tag{13}$$

This proves Equation (8), with $J \geq 0$ by nonnegativity of KL. The derivation applies to dependent mask coordinates and therefore to Soft Tail's fixed-budget law. It is the log-loss Bregman decomposition (Gruber & Buettner, 2023) under the effective view distribution q.

For logits h_z with $p_z = \mathrm{softmax}(h_z)$, the aggregate and disagreement have the forms

$$\bar{p} = \mathrm{softmax}(\mathbb{E}_q h_z), \qquad J = \mathbb{E}_q \, \mathrm{LSE}(h_z) - \mathrm{LSE}(\mathbb{E}_q h_z), \tag{14}$$

where $\mathrm{LSE}(h) = \log \sum_v e^{h_v}$. The second expression is the Jensen gap of log-sum-exp: one part of the loss measures consensus fit, and the other measures variation among corrupted-prefix predictions.

B.2 The view distribution induced by weighting

Using $w = (\beta + S)^{-1}$,

$$\mathbb{E}_{Q^w} S = \frac{\mathbb{E}_Q[S/(\beta + S)]}{a} = a^{-1} - \beta. \tag{15}$$

Convexity of $(\beta + u)^{-1}$ gives $a \geq (\beta + \mathbb{E}_Q S)^{-1}$ and therefore $\mathbb{E}_{Q^w} S \leq \mathbb{E}_Q S$. Also, $0 \leq S_n \leq 2/(1-\rho)$ and

$$\frac{1}{\beta + 2/(1-\rho)} \leq w_n \leq \frac{1}{\beta}. \tag{16}$$

Because the weight is bounded and strictly positive, the reweighted view distribution is well defined for every finite-length mask law.

B.3 Decision stability

Suppose $\bar{p}$ has a unique top token v^* with probability margin $\Delta > 0$ over the next most probable token. If a view predicts another token v, then

$$\Delta \leq \bar{p}(v^*) - \bar{p}(v) \leq |\bar{p}(v^*) - p_z(v^*)| + |\bar{p}(v) - p_z(v)| \leq \|\bar{p} - p_z\|_1. \tag{17}$$

Pinsker's inequality and Markov's inequality therefore imply

$$\Pr_q[\arg\max p_z \neq \arg\max \bar{p}] \leq \min\{1, 2J/\Delta^2\}. \tag{18}$$

This bound controls agreement with the consensus token. Consensus fit to the clean target and individual-view accuracy measure the correctness of those predictions.

B.4 SEPARATING POSITION AND MASK-DEPENDENT WEIGHTS

The experiments use a fixed positive profile b_n, stored under the field name `a_hat`. The four implemented weight rules have the following expected contributions:

token weight	mean weight	view law
1	1	Q
b_n	b_n	Q
w_n/b_n	a_n/b_n	Q_n^w
w_n	a_n	Q_n^w.

(19)

Comparing position-only with unit weights applies the multiplier b_n under view law Q. Comparing CARD with position-normalized weights applies that multiplier under Q_n^w. The expected weight a_n describes the corruption law, whereas b_n is the frozen calibration estimate. When $b_n = a_n$, position normalization gives an exact mean weight of one.

We estimate the profile from 50,000 corruption draws, check it on a disjoint 10,000-draw bank, and store it in float32. Every run uses this fixed profile throughout training.

B.5 SHARED-CORRUPTION MEASUREMENTS

The common diagnostic bank contains 2,048 targets from 512 held-out segments, sampled at positions 128, 576, 1024, and 1472. All seven models are evaluated on the same target IDs and mask views. The 32-view bank is a subset of the 64-view bank.

For M views, set $\pi_j = w_j / \sum_k w_k$ and $\widehat{a} = M^{-1} \sum_j w_j$. The empirical consensus is $\widehat{\bar{p}}(v) \propto \exp(\sum_j \pi_j \log p_j(v))$, giving target-level weighted loss $\widehat{a}(F_w + J_w)$. We average this quantity over targets. Here $\widehat{a}$ is estimated from the diagnostic views, whereas b_n is the fixed training profile. Diagnostic clean NLL is computed only on these selected targets and is therefore distinct from full-corpus clean-prefix PPL.

Table 18: Prediction on shared corrupted prefixes at 2,048 target positions. Loss and divergence are in nats. Disagreement compares view predictions with their consensus; accuracy compares them with the clean token.

Model	F_w	J_w	J_Q	Disagree (%)	Acc. (%)
AR	3.848	0.239	0.311	22.9	31.3
Uniform causal	3.498	0.183	0.244	18.7	34.3
Unit weights	3.347	0.154	0.212	16.7	35.9
Position-only weights	3.300	0.147	0.202	16.2	36.6
Position-normalized	3.279	0.129	0.190	14.9	37.2
Uniform + weights	3.336	0.143	0.201	16.0	36.5
CARD	3.205	0.111	0.170	13.7	38.0

Table 19: Diagnostic sensitivity to the number of shared views. The 32-view bank is nested in the 64-view bank.

Model	F_w, 32	F_w, 64	J_w, 32	J_w, 64
AR	3.848	3.847	0.239	0.232
Uniform causal	3.498	3.496	0.183	0.176
Unit weights	3.347	3.346	0.154	0.147
Position-only weights	3.300	3.299	0.147	0.141
Position-normalized	3.279	3.278	0.129	0.123
Uniform + weights	3.336	3.336	0.143	0.136
CARD	3.205	3.204	0.111	0.105

C Training Algorithm and Probabilistic Connections

Algorithm 1 CARD training for a length-L clean sequence

Require: $\mathbf{x}$, $\lambda \geq 1$, $\beta > 0$, $0 < \rho < 1$

1: Draw $t \sim \mathcal{U}(0,1)$ and set N, W by Equation (3).
2: Draw N mask positions uniformly from the final W positions.
3: Replace those tokens by `[MASK]` and prepend BOS, retaining L shifted inputs.
4: Compute all causal predictions for the clean targets in one forward pass.
5: Set $m_0 = 0$ and $S_1 = 0$.
6: **for** $n = 1, \ldots, L$ **do**
7: Set $w_n = (\beta + S_n)^{-1}$ and $\ell_n = -\log p_\theta(x_n \mid \mathbf{x}^z_{<n})$.
8: Set $S_{n+1} = \rho S_n + m_n(1 + m_{n-1})$.
9: **end for**
10: Update θ using $L^{-1} \sum_n w_n \ell_n$.

C.1 Independent absorbing corruption

Independent absorbing corruption connects dense causal supervision to discrete masked diffusion (Austin et al., 2021; Sahoo et al., 2024; Ou et al., 2025). On a time grid $0 = t_0 < \cdots < t_T = 1$, let $0 = \sigma_0 < \cdots < \sigma_T = 1$ and independently mask each coordinate with marginal probability σ_s. A Markov process realizing this marginal replaces an observed token by `[MASK]` between steps $s-1, s$ with probability $(\sigma_s - \sigma_{s-1})/(1 - \sigma_{s-1})$. For a masked coordinate, the posterior probability of having been observed at the previous step is

$$\eta_s = \frac{\sigma_s - \sigma_{s-1}}{\sigma_s}. \tag{20}$$

The reverse process copies observed tokens; for a masked token, it places mass $\eta_s p_\theta(v \mid \mathbf{x}^s_{<n})$ on clean token v and mass $1 - \eta_s$ on `[MASK]`. Reverse transitions factorize over coordinates conditional on the current noisy sequence. Starting from the all-mask prior defines a clean marginal $p^{\text{diff}}_{\theta,T}$.

Theorem C.1 (Dense causal variational bound)**.** *Under this independent absorbing process, the stated reverse parameterization, and finite logits,*

$$-\log p^{\text{diff}}_{\theta,T}(\mathbf{x}) \leq \sum_{s=1}^{T} (\sigma_s - \sigma_{s-1}) \sum_{n=1}^{L} \mathbb{E}[-\log p_\theta(x_n \mid \mathbf{x}^s_{<n})]. \tag{21}$$

As the grid mesh tends to zero, the right-hand side converges to

$$\int_0^1 \sum_n \mathbb{E}[-\log p_\theta(x_n \mid \mathbf{x}^t_{<n})]\, d\sigma(t). \tag{22}$$

Proof. The terminal KL term in the diffusion variational bound is zero. Observed coordinates have zero reverse-transition KL; a masked coordinate contributes $\eta_s[-\log p_\theta(x_n \mid \mathbf{x}^s_{<n})]$. The target-mask indicator is independent of its prefix loss under independent coordinate corruption. Taking its expectation contributes σ_s, and $\eta_s \sigma_s = \sigma_s - \sigma_{s-1}$, giving Equation (21). At fixed parameters and clean sequence, the prefix expectation is a finite sum over mask patterns with polynomial probabilities in σ. Finite logits make it finite and continuous, so the Riemann–Stieltjes sums converge to the stated integral. □

For linear corruption at fixed $t \in (0,1)$, let $X_n = -\log p_\theta(x_n \mid \mathbf{x}^t_{<n})$ and $Y_n = t^{-1}\mathbf{1}[x^t_n = \texttt{[MASK]}]X_n$. Conditional on the clean sequence,

$$\mathbb{E}[Y_n \mid \mathbf{x}^t_{<n}] = X_n, \qquad \text{Var}(Y_n) - \text{Var}(X_n) = \frac{1-t}{t}\mathbb{E}[X_n^2]. \tag{23}$$

Thus X_n is the Rao–Blackwellized version of the masked-target estimator Y_n: integrating out the target's own mask preserves the expectation and removes the variance term above. This argument establishes dense supervision for independent absorbing corruption. For CARD's dependent Soft Tail law, Eq. (8) characterizes the weighted objective directly.

C.2 CONTENT-INDEPENDENT CORRUPTION MIXTURES

For any content-independent corruption metadata z at fixed length L, define

$$p_\theta(\mathbf{x} \mid z, L) = \prod_n p_\theta(x_n \mid C_z(\mathbf{x}_{<n})). \tag{24}$$

Each factor is normalized and depends only on earlier tokens. Successive summation over $x_L, \ldots, x_1$ gives a normalized distribution, as does the mixture $p_\theta^{\text{mix}}(\mathbf{x} \mid L) = \mathbb{E}_z p_\theta(\mathbf{x} \mid z, L)$. Jensen's inequality yields

$$-\log p_\theta^{\text{mix}}(\mathbf{x} \mid L) \le \mathbb{E}_z \sum_n -\log p_\theta(x_n \mid C_z(\mathbf{x}_{<n})). \tag{25}$$

The unweighted objective therefore upper-bounds the negative log likelihood of a content-independent mixture of causal factorizations. CARD's weights additionally redistribute mass across targets and corruption views, as derived in Appendix B.4.

C.3 PREFIX DIVERSITY AND EXACT MASK SUPPORT

Corruption supplies multiple prediction contexts for each clean sequence. With unrestricted independent prefix masking, target n has 2^{n-1} possible masked histories, giving $2^L - 1$ target–context pairs across a length-L sequence. AR uses one clean history per target. Soft Tail selects a structured subset through its mask budget and eligible suffix.

For a fixed mask budget N and tail length W, the first $L - W$ positions are observed. At target n, let $r = \max(0, n - 1 - (L - W))$ be the number of eligible prefix positions. A particular complete prefix mask pattern with b masks has probability

$$\Pr(\text{prefix pattern} \mid N, W, n) = \frac{\binom{W-r}{N-b}}{\binom{W}{N}}. \tag{26}$$

The numerator chooses the remaining masks outside the prefix; infeasible binomial coefficients give probability zero. A specified set of k eligible positions being masked, without fixing the other positions, has probability $\binom{W-k}{N-k}/\binom{W}{N}$. The number of admissible complete prefix patterns is

$$\sum_{b=\max\{0,N-(W-r)\}}^{\min\{N,r\}} \binom{r}{b}. \tag{27}$$

The expected number of visible predecessors is

$$\min\{n-1, L-W\} + r\left(1 - \frac{N}{W}\right). \tag{28}$$

This expression separates the clean prefix from the visible tokens inside the tail window. It formalizes the context-preservation rationale for expanding the window beyond the mask budget.

For example, $L = 12, N = 4, W = 6$ permits masks at positions 8–11 with tokens 7 and 12 observed. The prediction for target 12 sees four preceding masks, since its own observed value is excluded by the shift. That complete pattern has probability $1/\binom{6}{4} = 1/15$.

Table 20: Soft Tail probabilities at $L = 2048$, averaged over uniform corruption time and targets 1921–2048. Local events mask the preceding $K - 1$ positions; full-prefix events also require the earlier prefix to be observed.

K	Local event	Full-prefix event	Local mass from $W = L$ (%)
8	0.157	2.45×10^{-5}	76.4
16	0.063	5.21×10^{-7}	97.9
32	0.031	1.21×10^{-10}	100.0

The reported marginals average targets 1921–2048 and uniform corruption time at $L = 2048$. Budget 1 has probability $2/L$; each budget from 2 through $L - 1$ has probability $1/L$. This includes saturated windows. These probabilities describe the mask patterns sampled during training. At inference, the visible prefix contains generated tokens and the unresolved block supplies its mask pattern.

D Additional Training and Likelihood Measurements

D.1 Independent model-family runs

The independent model-family study trains approximately 1B-parameter AR, MDLM, BD3LM, and CARD models on 300B FineWeb tokens at context length 2048. Each model uses its own training recipe. This study supplies the domain-likelihood and sequence-length measurements below; the main study supplies the initialization-paired comparisons. Both rank the models AR, CARD, MDLM, and BD3LM by downstream accuracy.

Table 21: Independent model-family comparison at approximately 1B parameters and 300B training tokens. Each model follows its model-specific training recipe; all eight tasks contribute equally to the average.

Model	ARC-C	ARC-E	CSQA	HellaS.	MMLU-r	PIQA	SciQ	Wino.	Mean
AR	34.04	64.65	52.74	61.26	25.45	75.95	81.10	55.96	56.39
MDLM	29.44	49.16	36.45	48.32	26.36	59.63	76.60	54.46	47.55
BD3LM	27.30	48.06	44.06	42.48	26.93	59.79	79.60	51.38	47.45
CARD	32.68	60.77	48.73	53.29	25.65	71.71	79.80	53.28	53.24

Table 22: Training-step benchmark for the 1B model-family implementations: eight H800 GPUs, batch 4 per device, accumulation 16, maximum context 2048, and 500 post-warmup steps.

Model	Seconds/step	Time / CARD
CARD	2.19	1.00×
MDLM	3.31	1.51×
BD3LM	6.58	3.00×

Table 23: Length sweep of the 1B model-family implementations (seconds per step). The four sequence lengths are measured in the same sweep.

Model	512	1024	2048	4096
CARD	1.44	1.73	2.57	4.78
MDLM	2.06	2.63	4.05	7.48
BD3LM	3.59	4.61	7.46	13.99

The fixed-context benchmark uses eight H800 GPUs, batch 4 per device, accumulation 16, and 500 post-warmup updates at length 2048. A separate length sweep evaluates 512, 1024, 2048, and 4096 tokens under a common sweep configuration. At length 4096, CARD takes 4.78 seconds per step and BD3LM takes 13.99 seconds, a 2.9-fold difference.

Table 24: Domain likelihood in the independent 1B model-family study. AR and CARD use exact clean-prefix PPL; MDLM and BD3LM use exponentiated variational NLL. The last column is the arithmetic mean over domains.

Model	AG News	arXiv	LAMB.	LM1B	OWT	PTB	PubMed	Wiki.	Mean
Exact clean-prefix PPL									
AR	30.62	18.15	33.83	39.14	17.68	117.56	11.93	40.52	38.68
CARD	27.67	20.34	30.36	29.61	17.59	97.74	13.20	38.67	34.40
Exponentiated Monte Carlo variational NLL									
MDLM	42.20	23.58	35.87	48.68	20.77	168.19	17.23	42.18	49.84
BD3LM	41.18	44.60	39.17	40.04	40.97	118.30	34.66	39.28	49.78

Table 24 reports exact clean-prefix PPL for AR and CARD and exponentiated Monte Carlo variational NLL for MDLM and BD3LM. CARD improves over AR on six of eight domains, reducing LM1B PPL from 39.14 to 29.61, LAMBADA from 33.83 to 30.36, and PTB from 117.56 to 97.74. AR leads on arXiv and PubMed. The arithmetic mean over domains is 38.68 for AR and 34.40 for CARD; these values average domain PPLs rather than pooling token losses.

D.2 LM1B LIKELIHOOD AND TRAINING TRAJECTORIES

Table 25: LM1B with 110M models, 33B training tokens, context 128 and matched EMA. The table labels each evaluation quantity.

Model	Evaluation	Value
AR	Exact clean-prefix	21.12
BD3LM	MC variational	35.06
MDLM	MC variational	37.48
CARD	Exact clean-prefix	21.54

The 110M LM1B likelihood study uses 33B training tokens, context length 128, and matched exponential moving average (EMA) handling. AR obtains exact clean-prefix PPL 21.12 and CARD 21.54; MDLM and BD3LM use Monte Carlo variational estimates (Table 25). This is a separate training regime from the 3B-token, context-2048 component study.

Table 26: AR and CARD in the 1B model-family study. Each model contributes one training trajectory. Mean domain clean-prefix PPL and task accuracy are evaluated at each checkpoint.

Tokens (B)	AR acc.	CARD acc.	Δ (pp)	AR PPL	CARD PPL
50	42.15	35.72	-6.43	63.80	60.40
100	48.02	42.62	-5.40	51.20	47.60
200	53.18	49.11	-4.07	43.20	38.90
300	56.39	53.24	-3.15	38.68	34.40

Table 26 follows the independent AR and CARD trajectories at four token budgets. CARD has lower mean domain clean-prefix PPL at every checkpoint, and its downstream accuracy deficit narrows from 6.4 to 3.2 points. The matched suite in Section 3 reproduces the direction of this likelihood–accuracy difference with three paired seeds.

D.3 REPEATED-DATA TRAINING

The fixed-data study trains 1B-parameter AR, MDLM, BD3LM, and CARD models for 40 passes over a 1B-token FineWeb-Edu subset (Penedo et al., 2024). HellaSwag accuracy tracks performance as the same corpus is reused. Table 27 gives the approximate crossover points: CARD exceeds AR near pass 12, MDLM exceeds AR near pass 23, and MDLM exceeds CARD near pass 27. CARD therefore reaches the crossover with AR earlier, while MDLM takes the lead later in training.

Table 27: Reported HellaSwag crossover points in the 40-pass experiment on a fixed 1B-token FineWeb-Edu subset. The experiment includes AR, MDLM, BD3LM and CARD.

Crossover	Approximate pass
CARD exceeds AR	12
MDLM exceeds AR	23
MDLM exceeds CARD	27

Prefix corruption presents a clean target under different partially observed histories on successive exposures. The support calculation in Appendix C describes this diversity, and the fixed-data results

measure how the model ordering changes over repeated training. These are distinct from the main suite's checkpoints along a 300B-token stream.

E Generation Protocol and Complete Results

E.1 Decoding rules and evaluation sets

All six decoders use the same `CARD-1B-main-final` checkpoint, tokenizer, and greedy token selection. Jacobi uses $K = 8$ and iterates to a fixed point without a preset cap. Lookahead uses level 5, window size 7, and guess-set size 7 with verified acceptance. Confidence-based CARD uses $(K, T_{\max}) = (8, 8), (16, 16), (32, 8)$ with $\tau = 0.9$.

For confidence-based decoding, let $\mathbf{c}$ be the completed prefix and let $\mathbf{y}^{(0)} = (\texttt{[MASK]}, \ldots, \texttt{[MASK]})$ be a length-K candidate block. At iteration j, compute

$$q_i^{(j)}(v) = p_\theta(v \mid \mathbf{c}, \mathbf{y}_{<i}^{(j)}), \qquad \widehat{v}_i^{(j)} = \arg\max_v q_i^{(j)}(v). \tag{29}$$

All positions are then updated simultaneously:

$$y_i^{(j+1)} = \begin{cases} y_i^{(j)}, & y_i^{(j)} \neq \texttt{[MASK]}, \\ \widehat{v}_i^{(j)}, & y_i^{(j)} = \texttt{[MASK]} \text{ and } q_i^{(j)}(\widehat{v}_i^{(j)}) > \tau, \\ \texttt{[MASK]}, & \text{otherwise.} \end{cases} \tag{30}$$

Committed tokens remain fixed while unresolved positions are recomputed. At the iteration cap, greedy predictions fill the remaining masks. The completed block is refreshed before its KV states are cached, so subsequent blocks use states computed from the completed causal prefix. The block size K, threshold τ, and cap $T_{\max}$ control decoding independently of the training sampler.

HellaSwag (Zellers et al., 2019) uses 1,000 validation prefixes. WikiText-103 (Merity et al., 2017) uses 1,000 test prefixes of 256 tokens, and PG-19 (Rae et al., 2020) uses 500 test prefixes of 512 tokens. Output caps are 128, 128, and 512 tokens. TriviaQA-unfiltered (Joshi et al., 2017) uses 2,000 validation questions with a fixed five-shot prompt and stops at EOS, a double newline, or 64 new tokens. Decoder hyperparameters are fixed across evaluation sets. The validation protocol defines a conservative operating point using an external-score ratio of at most 1.05, an F1 decrease of at most 0.5 points, a Rep-4 increase of at most 0.2 points, and an output-length change of at most 3%. The larger-block configurations evaluate more aggressive operating points.

E.2 Scoring

Let $\ell_{i,s}$ be scorer s's mean token NLL for output i, and let T_i count tokens under the generating model's tokenizer. The continuation score is

$$E_s = \exp\left(\frac{\sum_i T_i \ell_{i,s}}{\sum_i T_i}\right). \tag{31}$$

We use four external language models: Qwen3-8B (Qwen Team, 2025), SmolLM3-3B (Bakouch et al., 2025), Gemma-3-27B (Gemma Team, 2025), and GPT-2 Large (Radford et al., 2019). Each scorer evaluates the same generated continuations for all six decoders. The columns `qwen`, `smol`, `gemma`, and `gpt2` in the supplementary data correspond to these models, respectively.

Equation (31) uses each scorer's token NLL and weights samples by generated length T_i; this convention is identical across decoders. We compare E_s with the sequential reference under the same scorer and dataset, so positive relative changes mean worse continuation likelihood. Absolute values are reported separately for each scorer because their tokenizers differ. This generated-length weighting also distinguishes E_s from held-out corpus PPL, which divides a model's total NLL by its own evaluation-token count.

Rep-4 is the fraction of generated four-gram occurrences that repeat an earlier four-gram, aggregated across samples. Exact agreement compares full generated token arrays. Normalized token edit distance is Levenshtein distance divided by the longer output length. TriviaQA EM and F1 use the fixed reference-answer scorer in the evaluation package; questions and outputs are linked by their recorded identifiers and hashes.

Table 28: Continuation likelihood scores for all decoders. Scorer columns are E_s from Equation (31); Rep-4 is the percentage of repeated generated four-gram occurrences. Length counts generated tokens.

Decoder	Qwen3	SmolLM3	Gemma3	GPT-2 L	Rep-4 (%)	Length
HellaSwag						
Sequential	9.59	9.93	12.12	12.43	1.60	126.3
Jacobi	9.59	9.93	12.12	12.43	1.60	126.3
Lookahead	9.59	9.93	12.12	12.43	1.60	126.3
CARD-8	9.87	10.21	12.46	12.75	1.58	126.2
CARD-16	10.47	10.82	13.24	13.52	1.80	125.8
CARD-32	13.25	13.79	16.76	17.14	5.28	121.4
WikiText-103						
Sequential	12.67	13.08	15.60	16.18	1.67	125.2
Jacobi	12.67	13.08	15.60	16.18	1.67	125.2
Lookahead	12.67	13.08	15.60	16.18	1.67	125.2
CARD-8	13.07	13.47	16.13	16.66	1.66	125.2
CARD-16	13.99	14.47	17.25	17.88	1.85	124.7
CARD-32	17.30	17.91	21.26	22.06	5.26	121.1
PG-19						
Sequential	18.40	18.88	22.26	23.05	0.68	487.1
Jacobi	18.40	18.88	22.26	23.05	0.68	487.1
Lookahead	18.40	18.88	22.26	23.05	0.68	487.1
CARD-8	19.03	19.52	23.02	23.83	0.67	486.8
CARD-16	20.21	20.70	24.39	25.29	0.69	485.1
CARD-32	25.04	25.65	30.08	31.08	1.92	468.1

Table 29: TriviaQA with 2,000 fixed five-shot prompts. Agreement and normalized edit distance compare the output with sequential CARD on the same question.

Decoder	EM	F1	Agree (%)	Token edit	Length
Sequential	16.55	21.94	100.0	0.000	17.5
Jacobi	16.55	21.94	100.0	0.000	17.5
Lookahead	16.55	21.94	100.0	0.000	17.5
CARD-8	16.00	21.64	87.1	0.042	17.5
CARD-16	15.55	21.27	73.7	0.111	17.4
CARD-32	13.35	20.24	33.0	0.388	16.9

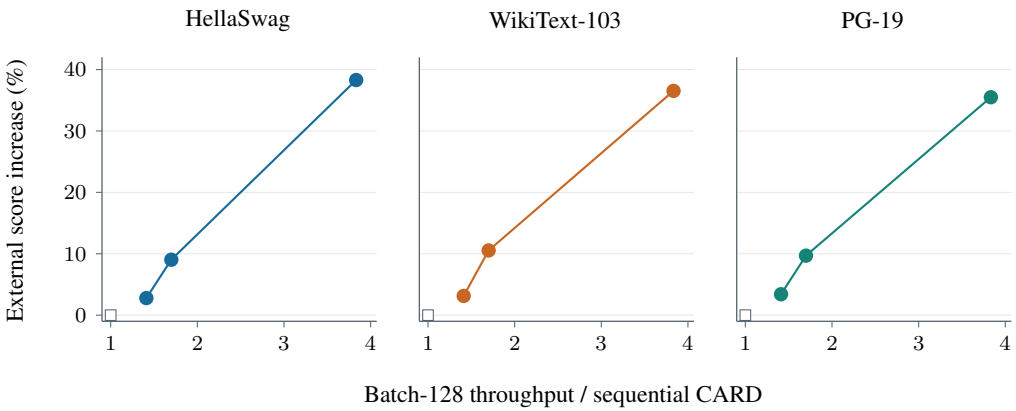


Figure 4: Continuation quality versus relative throughput. Lines connect CARD-8, CARD-16, and CARD-32. Whiskers span the four scorer-relative changes in E_s (positive is worse). Throughput is measured in the common batch-128 benchmark. The hollow square is the sequential reference.

E.3 Timing and computational work

The generation environment has eight NVIDIA H800 80GB HBM3 GPUs, driver 550.127.05, CUDA 12.4, PyTorch 2.5.1 and FlashAttention 2.7.4. Model and cache values use bfloat16; compilation uses `torch.compile(mode=max-autotune)`. Timing starts with prompts already resident on GPU, immediately before prefill, and ends after stop handling, candidate verification or state refresh, and output finalization. CUDA is synchronized at both boundaries. Model loading, compilation, ten warmup batches, and offline scoring are excluded.

For each decoder–batch-size combination, 20 timing windows record generated tokens per second; tables report their arithmetic mean and sample SD. Latency medians and 95th percentiles use 240 individual samples for each of the twelve decoder/batch groups. All 27,000 recorded output arrays enter the agreement, edit-distance, and repetition checks. Jacobi and Lookahead match the cached sequential arrays on all 4,500 prompts.

Table 30: Completion latency (seconds) and peak memory (GB). Each latency distribution contains 240 measurements at its stated batch size.

Decoder	p50, b1	p95, b1	p50, b128	p95, b128	GB, b1	GB, b128
Sequential	1.84	2.01	4.14	4.51	18.7	61.8
Jacobi	1.94	2.16	4.32	4.72	19.6	63.9
Lookahead	1.55	1.69	3.43	3.72	20.4	64.8
CARD-8	1.30	1.42	2.93	3.19	19.8	66.2
CARD-16	1.15	1.26	2.49	2.69	21.2	68.7
CARD-32	0.89	0.97	1.10	1.19	23.6	73.9

Table 31: Execution statistics at batch 128. Forced commitments occur at the cap, and refresh time is included in elapsed time. Lookahead proposes 1.71 candidates per generated token and accepts 1.22 verified tokens per forward pass.

Decoder	Passes/block	Forced (%)	Refresh (%)
Sequential	1.00	0.0	0.0
Jacobi	6.82	0.0	4.9
Lookahead	—	0.0	0.0
CARD-8	3.32	1.3	8.3
CARD-16	5.74	2.6	10.8
CARD-32	4.11	18.7	14.9