SIMPLE DIFFUSION LANGUAGE MODELS ARE MORE EFFECTIVE FEW-STEP GENERATORS THAN REPORTED

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models (DLMs) promise fast parallel generation, yet high-quality samples often require large number of refinement steps, which diminishes their advantage in practice. This has led to massive interest in and rapid development of new methods for effective few-step generation. We show that much of the supposed quality gap at few steps can instead arise from a suboptimally configured sampler. Modest sampler sharpening, without any model retraining, enables a couple years old masked DLM to rival supposedly far improved successors. This differently sampled DLM in fact achieves lower generative perplexity in just 16 steps than what its standard sampler obtains with 1024, while improving both judged quality and semantic diversity. We further show that conventional per-output metrics can fundamentally obscure these gains, since any optimal trade-off between two such metrics can be attained by a generator supported on at most two outputs. We subsequently introduce GROUPEVAL, which separately evaluates quality and across-output semantic diversity, and offers fresh insights including uncovering how $1.5-4.7\times$ perplexity gains of a distilled model yield no corresponding quality gain. Finally, we explain why sharpening helps: parallel unmasking destroys dependencies among simultaneously generated tokens, creating a gap between prediction and generation. We prove that pervasive temperature choice of one is generically suboptimal under parallel sampling even for an exact denoiser, and that worse predictions can yield better samples. Through these results, we argue for a broader evaluation principle of treating the deployed generator as the object of comparison, benchmarking it against tuned baselines, and assessing quality and diversity jointly and with more human-aligned measures.

1 INTRODUCTION

Diffusion language models promise fast text generation by denoising many tokens in parallel, yet their sample quality falters at few steps. Efforts to close this gap span distillation (Deschenaux & Gulcehre, 2025; Hayakawa et al., 2025; Sahoo et al., 2025; Zheng et al., 2026), new diffusion processes (Lou et al., 2024; von Rütte et al., 2025; Pynadath et al., 2026), and new samplers (Zheng et al., 2025; Liu et al., 2025; Kim et al., 2025a). However, there's a critical nuance that is often neglected: training evaluates a denoiser, while deployment runs a finite-step generator built from its predictions. Better predictions may not necessarily produce better samples at a fixed sampling budget. We study this gap in the original masked diffusion language model (MDLM) (Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2025). Its standard sampler uses temperature one and 64-bit floating-point noise (fp64), adopted after Zheng et al. (2025) showed that 32-bit (fp32) categorical sampling silently sharpens draws and made earlier comparisons unfair.

We find that much of the few-step gap lies in the sampler. Holding the well-trained, public MDLM checkpoint fixed and sampling it slightly sharper, at temperature .9 and in fp32 (MDLM$^\sharp$), gives lower generative perplexity (gPPL, perplexity under an external pretrained model) at 64 steps than the standard sampler reaches at 1,024: 41 versus 107, and 53 with temperature alone. Evaluated identically to its successors, MDLM$^\sharp$ at 64 steps beats the best result of distilled DUO (Sahoo et al., 2025) at *any* budget, and from 128 steps on it matches DiDi-Instruct (Zheng et al., 2026) (Figure 1a). The standard sampler can also reverse research conclusions at the small budgets where parallel generation is most valuable. A checkpoint trained $44\times$ longer, or a model with $5\times$ more parameters,

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.

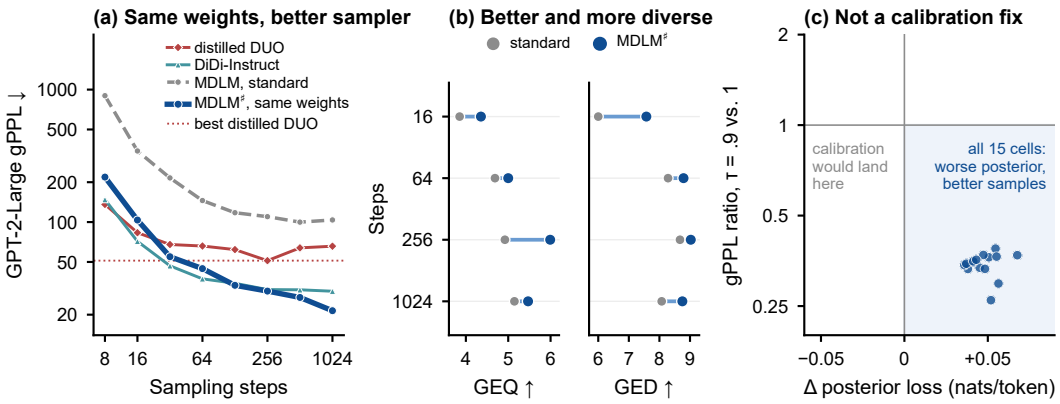


Figure 1: **Same weights, sharper sampler result in a better generator from a worse predictor. (a)** The unchanged public MDLM, sampled as MDLM$^\sharp$ (τ=.9, fp32) instead of the standard (τ=1, fp64), rivals distilled successors evaluated under one protocol. **(b)** Under the default GROUPEVAL judge, quality and semantic diversity *both* rise despite fall in token entropy. **(c)** Across three checkpoints and five budgets, sharpening to $\tau = .9$ worsens held-out posterior cross-entropy in all 15 cases (paired 95% intervals above zero) while cutting gPPL 2.6–3.8×. Section 4 explains the prediction–generation gap.

becomes a *worse* 2–8-step generator based on gPPL, even though the longer-trained checkpoint has better held-out posterior loss throughout (Section 2.1).

Temperature is not a new knob knob (Holtzman et al., 2020; Chang et al., 2022), and it has even been observed recently how it can substitute for steps along MDLM's perplexity–entropy frontier (Pynadath et al., 2026). But does a better point on that frontier mean a better *generator*? The standard metrics cannot tell. Perplexity and token entropy grade each output on its own, so neither notices when a generator keeps producing the same good output. And this blind spot is structural. We prove that every optimal trade-off between two per-output scores can be reached by a generator with at most two distinct outputs (Proposition 1). A stronger evaluator model does not help, because the problem lies in scoring outputs one at a time. What is needed is an evaluation that compares outputs with each other, and GROUPEVAL provides it. A blinded judge reads a group of outputs from one generator, scores each for quality, and sorts them by meaningfully distinct content, yielding GROUPEVAL QUALITY (GEQ) and GROUPEVAL DIVERSITY (GED). GEQ asks whether each output is good. GED asks whether the outputs say different things. With both scores, MDLM$^\sharp$ is better *and* more diverse across sampling budgets (Figure 1b). On matched outputs, two likelihood evaluators prefer DiDi-Instruct to MDLM by 1.5–4.7× in perplexity, while GROUPEVAL favors MDLM in quality and diversity. We find DiDi's samples to repeat fluent passages, a pattern that per-output scores need not penalize.

One may wonder why should a trained denoiser needs sharpening. Sharpening in fact *worsens* held-out posterior loss at every checkpoint and budget measured, even as samples improve (Figure 1c). In parallel unmasking, tokens revealed together are drawn independently, so their dependence is lost even when every prediction is exact. Asked to fill in a two-token city name in one round, the sampler can pair "New" with "Angeles". Our finite-step analysis makes this precise. It splits the sampler's error exactly into the model's prediction error, the dependence lost by drawing tokens together, and a credit from averaging over reveal orders that changes with temperature (Theorem 1). Because that credit moves, the temperature that is best for prediction is generally not the one that is best for generation. Even for an exact denoiser, temperature one is almost never optimal once tokens are revealed in parallel (Theorem 2), and in a two-token example sharpening makes the sampler exact while worsening its predictions (Proposition 2). Better predictors therefore need not be better few-step generators, and held-out loss alone cannot certify a sampler.

Our key contributions are as follows:

1. **A stronger baseline that changes comparisons.** Tuning only temperature and categorical precision lets an unchanged checkpoint by Sahoo et al. (2024) rival its distilled successors, and it reverses apparent few-step benefits of longer training and larger models (Section 2).

2. **A reusable group-level evaluation.** We prove that per-output metrics cannot detect collapse and introduce GEQ and GED, two separately reported scores under a fixed protocol. They confirm the sharpening gain and find no quality lead behind a 1.5–$4.7\times$ perplexity lead (Section 3).
3. **A theory of the prediction–generation gap.** An exact finite-step accounting of parallel unmasking shows that temperature one is generically suboptimal even for an exact denoiser, and that a worse predictor can be the better few-step generator (Section 4).

Relation to prior work. Temperature–step trade-offs in DLMs have been explored (Pynadath et al., 2026; Shah et al., 2026), and prior theory quantifies dependence lost in parallel draws (Lavenant & Zanella, 2025; Chen et al., 2026). Here the tuned baseline changes model rankings, while our accounting includes learned, tempered denoisers and path mixing. GROUPEVAL builds on semantic diversity and LLM judging (Kuhn et al., 2023; Zhang et al., 2025; Zheng et al., 2023) by fixing a group-level protocol for unconditional generation and testing where conventional metrics disagree. Section D develops these connections.

2 THE FEW-STEP GAP IS LARGELY A SAMPLING ARTIFACT

A generator is a checkpoint plus a sampler. A few-step MDLM generator is indexed by $(\theta, M, \tau, \kappa)$: checkpoint, sampling steps, temperature, and categorical precision. From an all-mask sequence, each of the M ancestral updates independently decides which masked positions to reveal and draws their tokens from the denoiser's tempered clean-token distribution

$$q_{\theta,\lambda}(v \mid z) = \frac{\exp\{\lambda\, r_\theta(v \mid z)\}}{\sum_u \exp\{\lambda\, r_\theta(u \mid z)\}}, \qquad \lambda = 1/\tau, \tag{1}$$

where r_θ are logits given the partially masked sequence z (revealed tokens are kept). The precision κ is that of the random numbers behind each Gumbel-max token draw, not of the network. In fp32 these numbers are too coarse to reach the extreme tail, so rare tokens are drawn too seldom. Zheng et al. (2025) showed that this lowers the effective temperature and recommended fp64, which became the default in later codebases. Because MDLM draws the mask-or-token outcome in the same categorical, precision also shifts when tokens are revealed, so fp32 is a distinct sharpening, not a relabeled temperature. Steps count configured updates M, which bound but do not measure network calls (Section A.1).

One checkpoint, one sharpening, frozen before any judging. We use the unchanged public MDLM-Small checkpoint (Sahoo et al., 2024), unconditional OpenWebText generation of 1,024 tokens, and 128 samples per cell, scored by GPT-2 gPPL and by the mean per-sample unigram (token-frequency) entropy H in nats. The *standard* sampler is $(\tau{=}1, \mathrm{fp64})$. MDLM$^\sharp$ is $(\tau{=}.9, \mathrm{fp32})$, chosen once from a sweep over $\tau \in \{.8, \ldots, 1.1\} \times \{\mathrm{fp32}, \mathrm{fp64}\}$ as the sharpest setting that keeps $H \geq 5$ at most budgets (a collapse screen) before any GROUPEVAL judgment, and then frozen for every budget, judge, and model. Weights, schedule, and algorithm never change.

Temperature beats a $16\times$ step budget, and precision compounds it. Temperature alone moves the baseline further than a $16\times$ larger step budget: at fixed fp64, 64 steps at $\tau = .9$ reach gPPL 53.1, against 107.1 for the standard sampler at 1,024 (Table 1). Returning to fp32 compounds the gain, to 41.3 at 64 steps and a $4.9\times$ reduction at 1,024. The ordering is the same for two further

Table 1: **Two sampling knobs compound: at 64 steps, MDLM$^\sharp$ decisively outperforms the standard sampler at 1,024 (bold).** GPT-2 gPPL (lower is better) with unigram entropy H in brackets, 128 samples per cell. Temperature and precision each help on their own.

Steps M	Standard $(1, \mathrm{fp64})$	$(1, \mathrm{fp32})$	$(.9, \mathrm{fp64})$	MDLM$^\sharp$ $(.9, \mathrm{fp32})$
16	357.6 [5.79]	338.7 [5.78]	109.8 [5.37]	102.2 [5.35]
64	154.9 [5.71]	107.6 [5.62]	53.1 [5.36]	**41.3** [5.27]
256	115.5 [5.66]	61.5 [5.45]	40.9 [5.31]	28.1 [5.13]
1024	**107.1** [5.63]	42.5 [5.33]	36.6 [5.26]	21.8 [5.01]

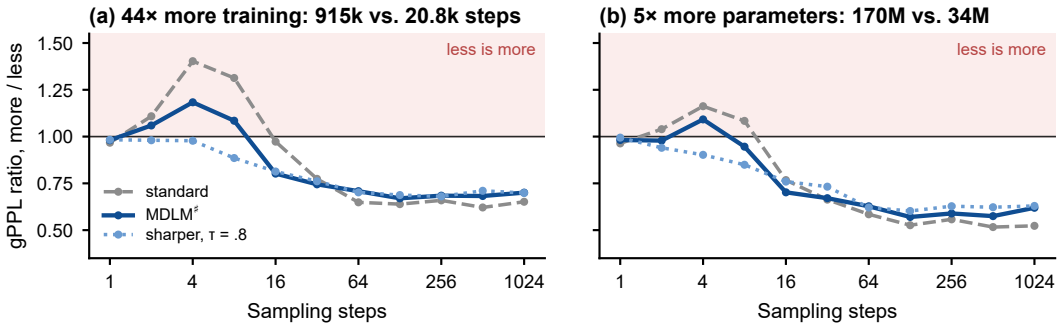


Figure 2: **At the standard sampler, more training and more parameters make few-step generation worse.** Ratio of gPPL for the larger or longer-trained model to the smaller or shorter-trained one, at matched steps and sampler. Values above one (shaded) mean that less is more. **(a)** MDLM-Small after 915k versus 20.8k training steps. **(b)** MDLM-Small versus MDLM-Tiny. Less wins only at 2–8 steps, and these inversions shrink under the sharpened sampler $(.9, \mathrm{fp32})$ and vanish with stronger sharpening. GPT-2 gPPL; 128 (Small) and 256 (Tiny) samples per cell.

MDLM-Small checkpoints from our own training run and for MDLM-Tiny (Sections A.2 to A.4). The fp32 draw is also cheaper: at 1,024 steps, MDLM$^\sharp$ uses 40% less peak memory and runs 17% faster than the standard sampler (Section A.5).

An unchanged baseline rivals its distilled successors. Figure 1a places MDLM$^\sharp$ among its successors under one protocol: GPT-2-Large gPPL on 40 samples per point, every other model at the standard sampler (Table 9). At 64 steps, MDLM$^\sharp$ scores 44.5, below the best result of distilled DUO at any budget (51.2) and the best of DUO (70.1) and SEDD (100.1). DiDi-Instruct keeps a perplexity lead through 64 steps. From 128 steps on, MDLM$^\sharp$ matches or beats it. Sharpening roughly halves DiDi's perplexity as well (Table 9), yet Section 3 finds that DiDi's perplexity lead is not a quality lead, and that its own sharpening brings no judge-robust quality gain.

2.1 Less is more, until the sampler is fixed

The standard sampler does not merely understate the baseline. It inverts comparisons that guide research decisions. We trained MDLM-Small with the reference recipe and kept checkpoints after 20.8k and 915k steps. At the standard sampler, the longer-trained checkpoint is the worse 2–8-step generator, by $1.40\times$ in gPPL at 4 steps (Figure 2a), even though its held-out posterior loss is lower at every audited budget, by .43–.62 nats per token (Section A.7). Model size shows the same inversion: the 170M-parameter MDLM-Small is a worse 2–8-step generator than a 34M-parameter MDLM-Tiny (Figure 2b). Both inversions shrink under the sharpened sampler and disappear with slightly stronger sharpening ($\tau = .8$). They are also confined to few steps: from 64 steps on, the larger or longer-trained model is 1.4–$1.9\times$ better in gPPL under all three samplers. A practitioner tuning few-step generation at the standard sampler would stop training early and choose the smaller model. Proposition 2 shows that such inversions need no optimization pathology: at a fixed temperature, a less accurate predictor can be the better few-step generator.

3 Are the Gains Real? Evaluate Groups, Not Samples

Lower perplexity with lower token entropy (Table 1) could also arise from a generator that repeats itself. We need to compare outputs to tell whether sharpening improves generation.

3.1 Per-sample metrics reward collapse

Generative perplexity and token entropy each grade one sample at a time and average the grades, so copying the best-graded output never costs anything. Formally, for length-L outputs under token-pooled scoring, $\log \mathrm{gPPL}(Q) = \mathbb{E}_{X\sim Q}[-\log R(X)]/L$ for an evaluator R, and the reported entropy

is $H(Q) = \mathbb{E}_{X\sim Q}\hat{H}(X)$, the mean unigram entropy of individual outputs. Both are linear in the generator Q, and linearity alone makes any protocol built from them degenerate.

Proposition 1 (Per-sample metrics reward collapse)**.** *Let f and g be any per-output scores on a finite output space. Every Pareto-optimal value of $(\mathbb{E}_Q f, \mathbb{E}_Q g)$ over generators Q (in particular, every solution of $\min_Q \mathbb{E}_Q f$ subject to $\mathbb{E}_Q g \geq h$) is attained by a generator supported on at most two outputs.*

The achievable pairs form the convex hull of per-output points, whose Pareto frontier consists of vertices and edges (Section B.1). Entropy-constrained perplexity minimization, the implicit target of perplexity–entropy reporting (Zheng et al., 2025; Pynadath et al., 2026), is therefore solved by repeating two fluent, lexically varied documents. A group of eight samples from such a generator holds at most two distinct outputs, which caps its GED (below) at 3.3 of 10. The MDLM samplers we study score 6–9. GROUPEVAL therefore asks directly whether a set of outputs contains good and genuinely different content.

3.2 GROUPEVAL: QUALITY PER OUTPUT, MODES PER GROUP

A judge sees a randomly ordered group of n outputs from one generator, with the generator, sampler, budget, and automatic scores hidden. Under a fixed rubric, it scores each output for readability, coherence, substance, and non-degeneration on $\{0, 1, 2\}$, after a gate that zeroes empty, corrupted, or non-communicative text. It also partitions valid outputs into semantic–discourse modes. Changes of topic, purpose, genre, or discourse structure create modes; paraphrases, entity swaps within a template, formatting, and corruption do not. Deterministic code converts these decisions into two complementary scores. **GROUPEVAL QUALITY (GEQ)** is the mean rubric total over *all* outputs, rescaled to $[0, 10]$, so failures count. **GROUPEVAL DIVERSITY (GED)** is validity-weighted, normalized mode entropy. For a group g of n outputs with V_g valid outputs and mode shares p_{gk},

$$\mathrm{GED}_g = 10\,\frac{V_g}{n}\,\frac{-\sum_k p_{gk}\log p_{gk}}{\log n}, \qquad \mathrm{GED} = \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}} \mathrm{GED}_g, \tag{2}$$

with $\mathrm{GED}_g = 0$ when fewer than two valid outputs or modes remain. Entropy rewards the number and balance of modes, and the validity factor charges for unusable text. With eight outputs (letters are modes, $\times$ is invalid text), the patterns AAAAAAAA, AAAAAAAB, AAAABBBB, and ABCDEFGH score 0, 1.81, 3.33, and 10; ABCD$\times\times\times\times$ scores 3.33. Merging modes or invalidating an output never raises GED (Section B.2). Valid but weak text keeps its mode mass, which is why GEQ accompanies GED. Read together, they separate the failure modes a single score would merge. Low GED with high GEQ flags good but similar outputs, and high GED with low GEQ flags varied but weak text.

A shared evaluation protocol. For unconditional generation we use groups of eight, the open-weight `gpt-oss-120b` judge, the fixed web-text rubric above, and two fixed partitions of each output pool. Every output is judged twice. GEQ intervals use a partition-blocked Bayesian bootstrap (Rubin, 1981). This blinded, automated procedure lets researchers compare checkpoints, samplers, and budgets under the same rubric without recruiting a new human panel for each comparison. We also conduct exploratory studies with four other judges to test robustness, and a task-specific conditional GEQ check for a separate rubric. These, alongside additional experimental details, are shared in Appendix B.

3.3 SHARPER SAMPLING IMPROVES QUALITY AND DIVERSITY TOGETHER

Under the default judge, MDLM$^\sharp$ raises GEQ and GED at every budget while token entropy falls by .44–.62 nats (Table 2A). Lower lexical entropy accompanies more distinct content. Both factors of GED improve at every budget: at 16 steps, the invalid share falls from 13.3% to 6.6% of outputs, and the effective number of modes among valid outputs per group of eight rises from 4.73 to 5.74. The quality gain is judge-robust. All five judges give MDLM$^\sharp$ the higher GEQ at all four budgets, 13 of these 20 conditional intervals exclude zero, and 16 of the 20 judge–budget pairs also show higher GED. The step savings survive judgment: MDLM$^\sharp$ at 256 steps beats the standard sampler at 1,024 on GEQ under every judge. Token entropy alone would have rejected this better generator.

Table 2: **GEQ and GED reveal what gPPL and token entropy miss.** Default judge; groups of eight. **(A)** Public MDLM, standard → MDLM$^\sharp$, 128 outputs per cell; GPT-2 gPPL. **(B)** MDLM minus DiDi-Instruct in a separate, matched, interleaved study, 40 outputs per model and cell; the same saved outputs receive GROUPEVAL and GPT-2-Large scores. gPPL ratios above one favor DiDi.

(A) Public MDLM-Small: standard → MDLM$^\sharp$

Steps	gPPL ↓	Token entropy H	GEQ ↑	GED ↑
16	357.6 → 102.2	5.79 → 5.35	3.85 → 4.36	6.00 → 7.57
64	154.9 → 41.3	5.71 → 5.27	4.69 → 5.00	8.28 → 8.79
256	115.5 → 28.1	5.66 → 5.13	4.92 → 5.99	8.67 → 9.02
1024	107.1 → 21.8	5.63 → 5.01	5.15 → 5.47	8.07 → 8.75

(B) MDLM vs. DiDi-Instruct on matched outputs

	Standard sampler			Sharpened sampler		
Steps	gPPL ratio	ΔGEQ	ΔGED	gPPL ratio	ΔGEQ	ΔGED
16	4.58×	0.00	+2.61	2.33×	+1.19	+4.65
64	3.78×	+0.20	+2.76	2.19×	+0.41	+1.79
256	3.49×	+1.50	+2.11	1.76×	+0.72	+3.04
1024	3.31×	+0.77	+2.61	1.48×	+1.06	+2.22

3.4 A PERPLEXITY LEAD IS NOT NECESSARILY A QUALITY LEAD

To compare model families on equal terms, we judge the same saved MDLM and DiDi-Instruct outputs behind Table 9 in a separate, matched, interleaved study: 40 outputs per model and cell. We rescore those outputs with two likelihood evaluators, GPT-2-Large and Qwen3-8B-Base (Yang et al., 2025), on the same token spans. The evaluators agree with each other (Spearman $\rho = .98$ across cells) and favor DiDi in all 8 comparisons, by 1.5–4.7×. GROUPEVAL's default judge disagrees (Table 2B): MDLM has the higher GEQ in 7 comparisons, ties in the eighth, and has the higher GED in all 8, often by a wide margin (6.6 versus 2.0 at 16 steps under the sharpened sampler). The one tie is telling: at 16 steps under the standard sampler, a 4.6× perplexity advantage buys DiDi no judged quality and costs it 2.6 points of GED. DiDi's outputs often repeat chunks of sensible sentences. A repeated chunk is fluent and predictable, which can lower perplexity, and it can still keep token entropy high. This qualitative observation suggests why the per-output metrics disagree with GROUPEVAL, while the measured disagreement survives first-document scoring and bits-per-byte (Section B.4.3).

GROUPEVAL distinguishes interventions and tracks task success. GROUPEVAL does not endorse every perplexity gain. Sharpening DiDi-Instruct roughly halves its gPPL but yields no judge-robust quality gain: every default-judge interval includes zero, and one robustness judge prefers the standard sampler (Table 12). With a separate task rubric and no access to answers or tests, conditional GEQ separates correct from incorrect LLaDA-8B and Dream-7B solutions with AUROC .91 on GSM8K and .96 on HumanEval, and follows the task-dependent ordering of the two models (Table 14).

4 WHY A PERFECT DENOISER NEEDS SHARPENING

Sections 2 and 3 establish that sharpening yields a better generator. Figure 1c adds a puzzle: the same sharpening worsens posterior prediction loss. Parallel sampling can produce this gap even from exact posteriors. The few-step generator is a *mixture over reveal paths*, each component a product of single-token predictions. Training shapes the components, while generation is judged on the mixture. Figure 3a draws it for two correlated bits and two rounds: half of the paths reveal both bits together and draw them independently, the other half reveal them in turn, and the output law averages the two branches.

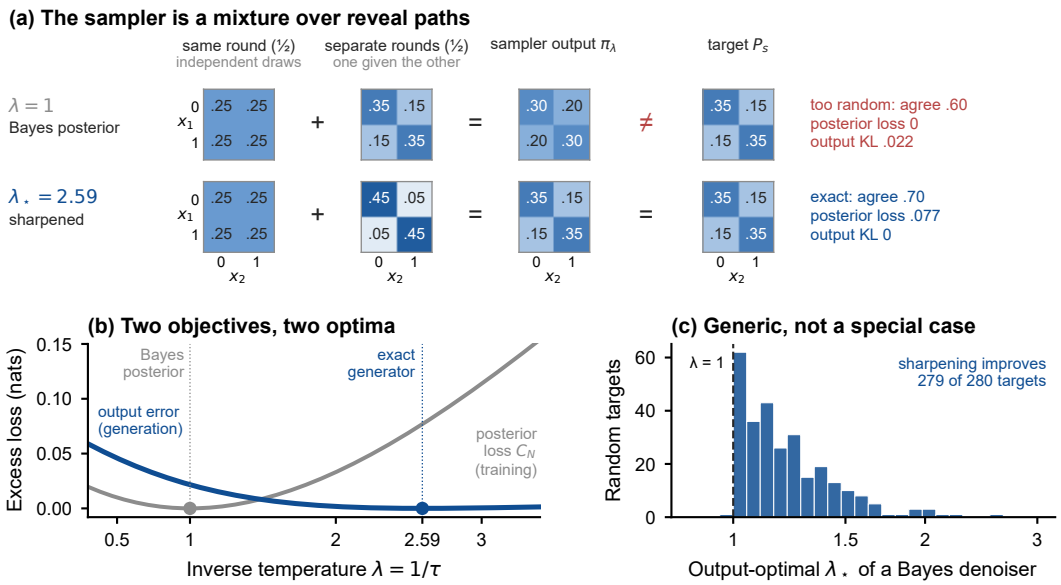


Figure 3: **Training and generation want different temperatures, even for a perfect model. (a)** The sampler as a mixture over reveal paths, for the pair family (6) with $s = .7$ and two equal rounds. Cells give the joint law of (x_1, x_2). At $\lambda = 1$, the independent same-round branch dilutes the exact sequential branch and the output is too random. At $\lambda_\star \approx 2.59$, the sequential branch over-agrees just enough to offset it. **(b)** The Bayes posterior minimizes posterior loss, but output error vanishes only at $\lambda_\star$. **(c)** Best inverse temperature for the *exact* Bayes denoiser on 280 random full-support targets ($L \in \{2, 3\}$, $|\mathcal{V}| \le 4$, random grids), by enumeration: never 1, and above 1 in all but one case.

4.1 Training integrates, sampling sums

Let $X \sim P$ on a finite space $\mathcal{V}^L$. A denoiser predicts a masked coordinate i from the visible values x_A and their positions, $q^i(\cdot \mid x_A)$, and $q^i_\lambda \propto (q^i)^\lambda$ is its tempered version. Its posterior-loss profile and the Bayes entropy profile are

$$c_\lambda(a) = \sum_{i=1}^{L} \mathbb{E}_{A,X}[-\log q^i_\lambda(X_i \mid X_A)], \qquad h(a) = \sum_{i=1}^{L} \mathbb{E}_A H(X_i \mid X_A), \tag{3}$$

where $A \subseteq [L] \setminus \{i\}$ retains each other coordinate independently with probability a. A sampler with grid $0 = a_0 < \cdots < a_N = 1$ gives each coordinate an independent reveal round K_i with $\Pr(K_i = k) = \delta_k := a_k - a_{k-1}$, which is exactly the law of MDLM's ancestral sampler (Section C.1). In round k, the coordinates $S_k = \{i : K_i = k\}$ are drawn independently from q_λ given all earlier rounds. Let ν be the law of the reveal path $S = (S_1, \ldots, S_N)$, $Q_\lambda(x, S)$ the joint law of output and path, and π_λ the output law. Predictions are positive and arithmetic is exact. Intuitively, training scores each token against contexts at every masking level, an area under the loss profile, whereas the sampler visits only N levels, a Riemann sum, and never lets same-round tokens see each other.

Theorem 1 (Training integrates, sampling sums)**.** *The continuous-time MDLM objective and the data entropy are areas under the two profiles, $\mathcal{L}_\infty(\lambda) = \int_0^1 c_\lambda(a)\, da$ and $H(X) = \int_0^1 h(a)\, da$. The few-step sampler evaluates their left Riemann sums instead. With $\mathcal{J}_N(\lambda) = \sum_k \delta_k c_\lambda(a_{k-1})$,*

$$\mathrm{KL}(P \otimes \nu \,\|\, Q_\lambda) = \mathcal{J}_N(\lambda) - H(X) = B_N + C_N(\lambda), \tag{4}$$

$$\mathrm{KL}(P \,\|\, \pi_\lambda) = B_N + C_N(\lambda) - R_N(\lambda), \tag{5}$$

where $B_N = \sum_k \delta_k h(a_{k-1}) - \int_0^1 h \ge 0$ is the expected conditional total correlation among tokens revealed in the same round, $C_N(\lambda) = \sum_k \delta_k [c_\lambda - h](a_{k-1}) \ge 0$ is the accumulated posterior error, and $R_N(\lambda) = \mathbb{E}_{X \sim P} \mathrm{KL}(\nu \,\|\, Q_\lambda(S \mid X)) \ge 0$ is the credit for mixing paths.

Conditioned on its reveal round k, a coordinate sees each other coordinate independently with probability a_{k-1}, so the path-augmented log-loss is exactly the left Riemann sum $\mathcal{J}_N$. Applying

the chain rule for KL in the other order, conditioning on the output first, peels off the path credit R_N (Section C.2). Each term has a distinct origin. C_N is posterior error: it vanishes for the Bayes denoiser at $\lambda = 1$, and it is what training and held-out loss measure. B_N is the price of parallelism. It depends only on the data and the grid, so a perfect model pays it in full. Because h is nonincreasing, it is the overshoot of a left Riemann sum, bounded by $\max_k \delta_k \sum_i I(X_i; X_{-i})$ and decaying like $1/N$ (Section C.2.3). Generation, however, is judged on the mixture, and the mixing credit R_N depends on λ. Minimizing posterior error therefore need not minimize output error. For two tokens the direction is explicit: an exact model whose tokens share a round with probability $c = \sum_k \delta_k^2$ has output entropy $H(\pi_1) \geq H(P) + c\, I(X_1; X_2)$, so parallel sampling makes a perfect model too random (Section C.2.3). Many steps do not remove the effect: with $M = L = 1024$, 63% of tokens still share their round with another token (Section C.2.3).

4.2 Temperature one is almost never right for parallel sampling

Theorem 2 (The Bayes temperature is generically suboptimal)**.** *Let q be the Bayes denoiser, $q^i(\cdot \mid x_A) = P(X_i = \cdot \mid X_A = x_A)$, with $L \geq 2$ and $|\mathcal{V}| \geq 2$. **(a)** If every round reveals at most one coordinate, in any order independent of X, then $\pi_1 = P$, so $\lambda = 1$ is optimal. **(b)** For the parallel sampler with any grid of $N \geq 2$ rounds, the full-support targets P for which $\lambda = 1$ is a stationary point of $\lambda \mapsto \mathrm{KL}(P \,\|\, \pi_\lambda)$ form a closed set of Lebesgue measure zero. For every other target, some $\lambda \neq 1$ gives the Bayes denoiser a strictly better output law.*

Part (a) is the chain rule. For (b), the derivative at $\lambda = 1$ is analytic in P, so its zero set is null (Mityagin, 2020) unless it vanishes identically, which a $|\mathcal{V}|$-ary pair family rules out (Section C.3).

The theorem separates parallel sampling from autoregressive decoding, where temperature one already gives the exact output law for an exact model. The theorem does not fix the direction, but the derivative has an explicit covariance form (Section C.3), and exhaustive enumeration finds sharpening to be the improving direction for 279 of 280 random small targets (Figure 3c). The pair bound of Section 4.1 suggests why: same-round draws add randomness that the target does not have, and sharpening removes it. Temperature is thus not a patch for a miscalibrated model. It is a free parameter of the parallel sampler that the Bayes posterior leaves unset.

4.3 Exact repair: a worse predictor can generate better

Two uniform bits that agree with probability s make the mechanism exact:

$$P_s(00) = P_s(11) = s/2, \qquad P_s(01) = P_s(10) = (1-s)/2, \qquad \tfrac{1}{2} < s < 1. \tag{6}$$

The tokens share a round with probability $c = \sum_k \delta_k^2$. That branch draws two independent uniform bits, which agree only half the time. Otherwise one token is revealed first and the other is drawn from its conditional, which the tempered Bayes denoiser makes agree with probability $u_\lambda = \mathrm{sigmoid}(\lambda \,\mathrm{logit}\, s)$.

Proposition 2 (Exact repair and inverted rankings)**.** *Let $0 < c < 1$. The sampled law is P_{t_λ} with $t_\lambda = c/2 + (1-c)u_\lambda$, and*

$$\mathrm{KL}(P_s \,\|\, \pi_\lambda) = \mathrm{KL}(\mathrm{Bern}(s) \,\|\, \mathrm{Bern}(t_\lambda)), \qquad C_N(\lambda) = (1-c)\,\mathrm{KL}(\mathrm{Bern}(s) \,\|\, \mathrm{Bern}(u_\lambda)). \tag{7}$$

(i) Exact repair. *If $c < 2(1-s)$, let $u_\star = (s - c/2)/(1-c)$. Then $\lambda_\star = \mathrm{logit}(u_\star)/\mathrm{logit}(s) > 1$ makes the sampler exact, $\pi_{\lambda_\star} = P_s$, although $C_N(\lambda_\star) > 0 = C_N(1)$. If $c \geq 2(1-s)$, output error decreases with every increase of λ.* ***(ii) Inverted ranking.*** *In the finite-repair regime, any denoiser with uniform marginals and conditional agreement $u \in (s, u_\star]$, sampled at $\lambda = 1$, has strictly larger posterior loss and strictly smaller output error than the Bayes denoiser.* ***(iii) Vanishing with steps.*** *As $c \to 0$, $\lambda_\star = 1 + \frac{s - 1/2}{s(1-s)\,\mathrm{logit}\, s}\, c + O(c^2)$, where $c = 1/N$ for N equal rounds.*

With two equal rounds and $s = .7$, the independent branch agrees with probability .5 and the sequential branch with .7, so the mixture agrees with probability .6: too random. Sharpening to $\lambda_\star \approx 2.59$ raises sequential agreement to .9, restoring the target .7 exactly (Figure 3a,b). In the accounting identity, C_N rises but R_N rises by more: improving the mixture requires worsening its components. Nothing here is special to temperature. Any continuous sharpening family that moves sequential agreement from s toward 1 passes through $u_\star$ and repairs the sampler; truncation-style

sharpening, including fp32 Gumbel noise, pulls the same lever. Part (ii) is the inversion of Section 2.1 in miniature: the less accurate predictor is the better few-step generator. Part (iii) explains why sharpening matters most at few steps and fades as rounds become sequential. The strict benefit survives small perturbations of the target and grid (Section C.4).

4.4 In trained models, sharpening hurts prediction and helps generation

If sharpening compensates for parallel sampling rather than for an underconfident model, it should improve samples while *worsening* the schedule-matched posterior loss $\mathcal{J}_N$ of Theorem 1. We estimate $\mathcal{J}_N$ without bias on 512 held-out OpenWebText sequences, masking one target token and drawing its context from the reveal grid (Section C.5.1). Moving from $\tau = 1$ to .9 raises $\mathcal{J}_{M+1}/L$ (M updates plus a final cleanup round) by .036–.068 nats per token for three MDLM-Small checkpoints at five budgets, with all 15 paired intervals above zero, and the posterior-optimal temperature on our grid is 1 or 1.05 (Table 10). Over the same cells, gPPL falls 2.6–3.8$\times$ (Figure 1c). The checkpoint inversion of Section 2.1 carries the same signature: the 915k-step checkpoint has lower $\mathcal{J}_N$ at every budget and a lower posterior-loss profile at every nontrivial clean-probability level (Section A.7), yet is the worse few-step generator at the standard sampler. Held-out loss sees only $B_N + C_N$; the samples also see R_N, so held-out loss cannot certify any sampler, ours included (Section D). The theory shows that the mechanism exists and is generic, not how the language-model gain splits among parallelization, evaluator preferences, and precision.

5 Discussion and Conclusion

We uncover how a masked diffusion language model from a few years ago, sampled slightly differently, rivals models distilled to replace it and gains quality and diversity together under group-level judging. Our findings have significant implications for the broader DLM landscape.

A new method should beat the baseline's best sampler. A gain over a default configuration mixes a new method's contribution with what the baseline sampler leaves on the table. Tune temperature and categorical precision for both under the same protocol, and report the search budget and the sampler behind each result. Sharpening can also be implicit, through categorical precision or a selection rule, and configured steps should be reported alongside network calls and measured costs.

Better predictions need not produce better generators. Training reduces posterior error but cannot restore the dependence lost when same-round tokens are drawn independently. Sharpening compensates for this loss, better schedules and joint draws reduce its source, and revision revisits committed tokens (Theorem 1 and Section D).

Generation quality requires comparisons across outputs. GEQ and GED report quality and across-output diversity side by side under one fixed protocol (Proposition 1 shows why both are needed). On outputs studied, they confirm MDLM's sharpening gain, find no judge-robust gain from sharpening DiDi-Instruct, and show that DiDi-Instruct's perplexity lead is not a judged quality lead.

Scope and limitations. Our central audit concerns unconditional generation with small masked diffusion models, albeit very popular ones. The cross-model comparison adds DUO, SEDD, and DiDi-Instruct, but tunes the sampler only for MDLM and DiDi-Instruct, so it does not show how far sharpening would move the other families. The tested larger 8B models already use greedy decoding, and positive temperatures do not help (Section A.8). The theory assumes exact, non-adaptive reveal sampling, and adaptive and revising samplers need further analysis. Comprehensive human validation of GroupEval's quality scores and mode assignments would add further value.

The sampler is part of the model. A checkpoint is only as good as the sampler it is deployed with, and its training loss cannot say which sampler that should be. Once the sampler is tuned, an old baseline stands with the methods built to replace it. Once outputs are compared with each other, a large perplexity lead can turn out to carry no quality lead. Both corrections come from measuring the deployed generator rather than its predictions or single samples. To identify progress in few-step generation, tune the sampler and evaluate the quality and range of what it produces.

REPRODUCIBILITY STATEMENT

Detailed proofs appear in Sections B.1 and C. We also share (anonymized version of) all relevant code, including precise prompts, alongside clear usage and reproduction guidelines.

ETHICS STATEMENT

This work evaluates text generation and does not involve a new human-subject study. Web-trained generators may produce harmful, biased, or false text. GroupEval measures rubric-based perceived quality and local semantic diversity, not factuality, safety, or demographic fairness. Judge biases and changes to hosted models can affect its scores. Our protocol preserves judge-specific results and records provenance rather than treating automated judgments as human ground truth.

AI USE STATEMENT

Generative AI tools assisted with literature retrieval, refinement of the theoretical framework, and creating presentation figures. The authors remain responsible for the final text and artifacts.

REFERENCES

Mansoor Ahmed, Yue-Tsz Fan, Hemanth Venkateswara, and Murray Patterson. Latent-kernel discrete flow maps for few-step generation. *arXiv preprint arXiv:2607.27529*, 2026. URL https://arxiv.org/abs/2607.27529.

Hasan Amin, Yuan Gao, Yaser Souri, Subhojit Som, Ming Yin, Rajiv Khanna, and Xia Song. Consistent diffusion language models. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/66178.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

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*, 2026. URL https://openreview.net/forum?id=XjcHRIu0iF.

Parikshit Bansal and Sujay Sanghavi. Enabling approximate joint sampling in diffusion LMs. *arXiv preprint arXiv:2509.22738*, 2025. URL https://arxiv.org/abs/2509.22738.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/510e22f4a2da5212a64f1591736e2eaf-Abstract-Conference.html.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b5b528767aa35f5b1a60fe0aaeca0563-Abstract.html.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Chang_MaskGIT_Masked_Generative_Image_Transformer_CVPR_2022_paper.html.

Mark Chen et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Sitan Chen, Kevin Cong, and Jerry Li. Optimal inference schedules for masked diffusion models. In *Proceedings of Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 1279–1311, 2026. URL https://proceedings.mlr.press/v336/chen26e.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Justin Deschenaux and Caglar Gulcehre. Beyond autoregression: Fast LLMs via self-distillation through time. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/0f93c3e9b557980d93016671acd94bd2-Abstract-Conference.html.

Justin Deschenaux, Caglar Gulcehre, and Subham Sekhar Sahoo. The diffusion duality, chapter II: Ψ-samplers and efficient curriculum. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/9adf58ca1ed848228377d61692d66541-Abstract-Conference.html.

Dan Friedman and Adji Bousso Dieng. The Vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=g97OHbQyk1.

Aaron Gokaslan and Vanya Cohen. OpenWebText corpus, 2019. URL https://skylion007.github.io/OpenWebTextCorpus/.

Samson Gourevitch, Yazid Janati, Dario Shariatian, Umut Simsekli, Eric Moulines, Eric P. Xing, and Alain Durmus. Uniform diffusion models revisited: Leave-one-out denoiser and absorbing state reformulation. *arXiv preprint arXiv:2605.22765*, 2026. URL https://arxiv.org/abs/2605.22765.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. Unifying human and statistical evaluation for natural language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1689–1701, 2019. doi: 10.18653/v1/N19-1169. URL https://aclanthology.org/N19-1169/.

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. URL https://proceedings.mlr.press/v267/hayakawa25a.html.

Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Demystifying MaskGIT sampler and beyond: Adaptive order selection in masked diffusion. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=mKlW68i2Ig.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *International Conference on Machine Learning*, 2025a. URL https://proceedings.mlr.press/v267/kim25ah.html.

Seo Hyun Kim, Sunwoo Hong, Hojung Jung, Youngrok Park, and Se-Young Yun. KLASS: KL-guided fast inference in masked diffusion models. In *Advances in Neural Information Processing Systems*, 2025b. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8514c1f12dcdd061ac8521feb78f28ca-Abstract-Conference.html.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Hugo Lavenant and Giacomo Zanella. Error bounds and optimal schedules for masked diffusions with factorized approximations. *arXiv preprint arXiv:2510.25544*, 2025. URL https://arxiv.org/abs/2510.25544.

Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gómez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/cbf883a744952d4a40591271a58ab9d0-Abstract-Conference.html.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, 2023. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/lou24a.html. Oral.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs created equal? A large-scale study. In *Advances in Neural Information Processing Systems*, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/e46de7e1bcaaced9a54f1e9d0d2f800d-Abstract.html.

Omer Luxembourg, Haim Permuter, and Eliya Nachmani. Plan for speed: Dilated scheduling for masked diffusion language models. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65445.

Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ByJHuTgA-.

Boris S. Mityagin. The zero set of a real analytic function. *Mathematical Notes*, 107(3):529–530, 2020. doi: 10.1134/S0001434620030189. URL https://link.springer.com/article/10.1134/S0001434620030189.

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*, 2025. 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 *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a365e37c18fb91af547a2f0012a89e98-Abstract-Conference.html.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/260c2432a0eecc28ce03c10dadc078a4-Abstract.html.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large-Margin Classifiers*. MIT Press, 2000. URL https://mitpress.mit.edu/9780262194488/advances-in-large-margin-classifiers/.

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. CANDI: Hybrid discrete-continuous diffusion models. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/66818.

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.pdf.

Donald B. Rubin. The Bayesian bootstrap. *The Annals of Statistics*, 9(1):130–134, 1981. doi: 10.1214/aos/1176345338. URL https://doi.org/10.1214/aos/1176345338.

Yusuf Sahin, Ahmed Rockey Saikia, Volkan Cevher, and Paolo Favaro. Attention-discounted adaptive sampler for masked diffusion language models. *arXiv preprint arXiv:2606.10829*, 2026. URL https://arxiv.org/abs/2606.10829.

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 *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T. Chiu, and Volodymyr Kuleshov. The diffusion duality. In *International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/sahoo25a.html.

Yash Shah, Abhijit Chakraborty, and Vivek Gupta. CaRE compute-aware remasking evaluation protocol for masked diffusion language models. *arXiv preprint arXiv:2607.24763*, 2026. URL https://arxiv.org/abs/2607.24763.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bad233b9849f019aead5e5cc60cef70f-Abstract-Conference.html.

Luhan Tang, Longxuan Yu, Shaorong Zhang, and Greg Ver Steeg. Is your diffusion sampler actually correct? a sampler-centric evaluation of discrete diffusion language models. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/61598.

Guy Tevet and Jonathan Berant. Evaluating the evaluation of diversity in natural language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 326–346, 2021. doi: 10.18653/v1/2021.eacl-main.25. URL https://aclanthology.org/2021.eacl-main.25/.

Gilad Turok, Yair Schiff, Christopher De Sa, and Volodymyr Kuleshov. DUEL: Exact likelihood for masked diffusion via deterministic unmasking. In *ICML Workshop on Foundations of Deep Generative Models*, 2026. URL https://openreview.net/forum?id=FQA0rTzTVd.

Dimitri von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. In *International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/von-rutte25a.html.

Martin J. Wainwright. The data geometry of masking diffusion: Certified-optimal schedules via unmasking growth complexity. *arXiv preprint arXiv:2608.13520*, 2026. URL https://arxiv.org/abs/2608.13520.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/d877ea0dd78bdbe54830670618c1de09-Abstract-Conference.html.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 9440–9450, 2024. doi: 10.18653/v1/2024.acl-long.511. URL https://aclanthology.org/2024.acl-long.511/.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dLLM: Training-free acceleration of diffusion LLM by enabling KV cache and parallel decoding. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5d8d4e6061c3ba96c240b7fa1ae3471d-Abstract-Conference.html.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

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. URL https://arxiv.org/abs/2508.15487.

Shaorong Zhang, Longxuan Yu, Rob Brekelmans, Luhan Tang, Salman Asif, and Greg Ver Steeg. Generation order and parallel decoding in masked diffusion models: An information-theoretic perspective. *arXiv preprint arXiv:2602.00286*, 2026. URL https://arxiv.org/abs/2602.00286.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. NoveltyBench: Evaluating language models for humanlike diversity. *arXiv preprint arXiv:2504.05228*, 2025. URL https://arxiv.org/abs/2504.05228.

Haoyang Zheng, Xinyang Liu, Cindy Xiangrui Kong, Nan Jiang, Zheyuan Hu, Weijian Luo, Wei Deng, and Guang Lin. Ultra-fast language generation via discrete diffusion divergence instruct. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/323880c576dc7cdcb1fcc7432447a3df-Abstract-Conference.html.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9e3b203e72c4e058de26d02a92a81844-Abstract-Conference.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

A Additional Experiments and Implementation Details

A.1 Experimental setup

This subsection consolidates dataset, model, sampler, and evaluation details for the reported experiments. Throughout, the deployed object is the sampler–model pair $\pi_0^{\theta,M,\tau,\kappa}$, where θ is the denoiser checkpoint, M is the configured number of reverse-update invocations, τ is the explicit sampling temperature, and $\kappa \in \{\mathrm{fp32}, \mathrm{fp64}\}$ is the categorical-kernel precision. Implementations also apply their configured terminal noise removal. Because ancestral caching can reuse logits on empty updates, actual model-call counts are path-dependent and were not instrumented in the quality grids. Note that M is not NFE, and cross-model rows are not compute-matched (the exact update/NFE relationship is derived in Section C.5.2). We call $(\tau{=}1, \mathrm{fp64})$ the *standard* sampler and $(\tau{=}0.9, \mathrm{fp32})$ the *sharpened* sampler; when applied to the public MDLM-Small checkpoint, the latter is MDLM$^\sharp$. Released aggregates label the two settings `reference` and `selected`. The sharpened sampler is an empirical, guard-constrained operating point, instead of a universal recommendation or a posterior-calibration optimum.

Data. The small-model unconditional-generation experiments use OpenWebText (Gokaslan & Cohen, 2019) with the GPT-2 BPE tokenizer (Radford et al., 2019). Unless stated otherwise, generation is unconditional at sequence length $L{=}1024$ tokens. The 128-sample MDLM grids use GPT-2 as the external evaluator. A controlled exact-output sensitivity study additionally applies GPT-2-Large and Qwen3-8B-Base evaluators under common span semantics (Table 13).

Models. We train two MDLM variants from the official codebase of Sahoo et al. (2024), denoted MDLM-Tiny and MDLM-Small, on OpenWebText with the standard SUBS parameterization, absorbing forward process, and continuous-time NELBO objective (Section 4.1); training hyperparameters follow Sahoo et al. (2024) unless noted. MDLM-Small matches the architecture and parameter count of the publicly released MDLM-Small checkpoint of Sahoo et al. (2024); MDLM-Tiny is a smaller variant retaining the same architecture family. The reported MDLM-Small early and late checkpoint identifiers are `0-20788.ckpt` and `8-914672.ckpt`, corresponding to near-completion of first and ninth epoch. The *public* checkpoint is the much longer trained MDLM-Small release of Sahoo et al. (2024), used as-is.

Cross-model checkpoints. For Table 9, one consolidated artifact contains 40-sample GPT-2-Large evaluations of released checkpoints for MDLM (Sahoo et al., 2024), DUO and DUO-DCD (Sahoo et al., 2025), SEDD (Lou et al., 2024), and DiDi-Instruct (Zheng et al., 2026). DiDi-Instruct uses the EMA student with its reward-guided ancestral sampler in the main-text comparison, while an unguided ancestral grid is also reported in Table 7.

Sampler. We use the MDLM ancestral sampler with a linear α-schedule and uniform time grid. Temperature is applied as post-hoc rescaling of the per-position x_0 marginals ((1)). $\kappa \in \{\mathrm{fp32}, \mathrm{fp64}\}$ selects the floating-point precision of the Gumbel-max categorical draw. We sweep $\tau \in \{0.80, 0.85, 0.90, 0.95, 1.00, 1.05, 1.10\}$ and $\kappa \in \{\mathrm{fp32}, \mathrm{fp64}\}$, at configured update budgets $M \in \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$.

Evaluation metrics. Likelihood-based evaluation uses generative perplexity (gPPL, lower is better), with the causal-LM evaluator identified for each experiment. The compact aggregates pair it with within-sequence empirical unigram entropy H in nats. We use a declared conservative operating guard $H \geq 5$: cells below it are marked with † and treated as aggressive Pareto points. This statistic exposes obvious within-sequence concentration but does not measure across-sample support coverage and is not a distributional-correctness certificate.

Hardware. Local generation and sampling-cost measurements use a single NVIDIA H100 (80 GB) GPU. GroupEval judges are documented separately. The matching cost comparison in Figure 5b uses batch size 8, one warm-up batch, and three measured batches (24 sequences) per point under one software environment.

Reproducibility. Sampling temperature and categorical-kernel precision are the only knobs varied between standard and sharpened rows. Model weights, time grids, and the reveal rule are not changed at all. We do not retrain a model, fine-tune a sampler, or distill a student whatsoever.

A.2 THE FOUR-CORNER AUDIT HOLDS ACROSS CHECKPOINTS AND BUDGETS

Table 3 extends Table 1 to every budget and to our two MDLM-Small checkpoints.

Table 3: **A stronger unchanged baseline across checkpoints and budgets.** Three MDLM-Small checkpoints, 128 unconditional OpenWebText samples per cell; entries are GPT-2 gPPL with unigram entropy H in brackets. The first four columns of settings form the four-corner audit; the last shows additional sharpening. Underlined: standard sampler. **Bold**: best of the four corners based on both gPPL and H. †: $H < 5$. The standard sampler has the largest gPPL in every listed checkpoint–budget cell; $(\tau = .9, \text{fp32})$ is best among the four corners for every $M \geq 2$. At $M = 1024$ it improves public-checkpoint gPPL by $4.9\times$ while passing the guard. M counts configured updates, not NFE.

M	$(1, \text{fp64})$	$(1, \text{fp32})$	$(.9, \text{fp64})$	$(.9, \text{fp32})$	$(.85, \text{fp32})$
(a) Public MDLM-Small (Sahoo et al., 2024) (the canonical baseline).					
1	5043.3 [5.83]	5001.7 [5.83]	**2091.2** [5.27]	2148.8 [5.26]	1306.6† [4.91]
2	3893.6 [5.93]	3725.3 [5.93]	1293.3 [5.29]	**1224.3** [5.27]	675.8† [4.84]
8	945.1 [5.89]	853.9 [5.87]	247.2 [5.34]	**235.5** [5.31]	131.6† [4.98]
32	196.8 [5.73]	158.7 [5.70]	66.0 [5.35]	**58.0** [5.34]	40.0 [5.13]
128	120.2 [5.67]	80.2 [5.55]	43.7 [5.33]	**32.3** [5.21]	22.9 [5.01]
512	105.9 [5.64]	50.0 [5.40]	38.4 [5.28]	**24.7** [5.06]	18.3† [4.88]
1024	107.1 [5.63]	42.5 [5.33]	36.6 [5.26]	**21.8** [5.01]	16.3† [4.82]
2048	106.5 [5.63]	32.3 [5.18]	38.6 [5.28]	**18.8**† [4.89]	14.4† [4.66]
(b) MDLM-Small late (this work).					
1	4779.7 [5.82]	4732.8 [5.82]	**1992.2** [5.25]	2044.9 [5.25]	1248.8† [4.90]
2	3634.1 [5.93]	3513.4 [5.92]	1280.5 [5.31]	**1219.4** [5.29]	704.9† [4.90]
8	795.6 [5.88]	725.3 [5.84]	236.4 [5.35]	**231.8** [5.32]	132.1 [5.00]
32	194.5 [5.72]	151.2 [5.68]	64.7 [5.35]	**60.9** [5.34]	40.7 [5.14]
128	121.4 [5.67]	80.1 [5.55]	44.3 [5.31]	**33.8** [5.20]	24.0 [5.00]
512	111.9 [5.64]	50.1 [5.38]	38.0 [5.26]	**24.7** [5.05]	18.0† [4.84]
1024	108.2 [5.63]	41.9 [5.33]	38.6 [5.26]	**22.5** [5.03]	16.7† [4.82]
2048	107.2 [5.62]	34.5 [5.19]	41.0 [5.27]	**19.2**† [4.87]	14.7† [4.70]
(c) MDLM-Small early (this work).					
1	4941.5 [5.80]	4864.3 [5.80]	**2006.8** [5.22]	2085.2 [5.23]	1280.6† [4.87]
2	3278.0 [5.89]	3104.6 [5.88]	1212.2 [5.31]	**1151.1** [5.29]	686.8† [4.93]
8	605.9 [5.80]	551.5 [5.78]	235.2 [5.41]	**213.7** [5.39]	138.6 [5.15]
32	251.2 [5.70]	205.1 [5.65]	88.1 [5.34]	**81.7** [5.32]	53.6 [5.13]
128	190.0 [5.67]	120.6 [5.53]	70.2 [5.30]	**50.4** [5.17]	34.8† [4.99]
512	180.1 [5.64]	75.3 [5.35]	60.1 [5.25]	**36.2** [5.02]	26.7† [4.84]
1024	166.1 [5.62]	64.4 [5.30]	57.5 [5.23]	**32.1**† [4.95]	24.0† [4.79]
2048	172.6 [5.63]	50.5 [5.18]	59.5 [5.26]	**26.9**† [4.86]	21.0† [4.70]

A.3 MDLM-TINY: THE SAME EFFECT AT SMALLER SCALE

Figure 4 reports the corresponding operating-point comparison for MDLM-Tiny. The same standard-versus-sharpened contrast holds: at every M, the sharpened cell improves gPPL by a substantial

multiple, with $H \geq 5$ over most of the measured budget range. The full MDLM-Tiny grid is in Table 4.

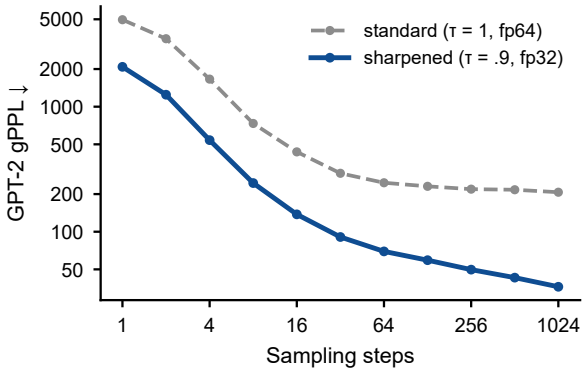


Figure 4: **MDLM-Tiny shows the same sampler gap at smaller scale.** Late MDLM-Tiny checkpoint; standard (τ=1, fp64), dashed; sharpened (τ=.9, fp32), solid. The gap mirrors the MDLM-Small headline (Table 1).

A.4 Full sampling-configuration grids

In each grid below, columns are the seven temperatures $\tau \in \{0.80, \ldots, 1.10\}$ and rows are configured reverse updates M. Both sub-blocks are shown (fp64 above, fp32 below). To aid scanning, the standard (τ=1, fp64) and sharpened (τ=0.9, fp32) cells are emphasized in **bold**. Cells below the $H < 5$ operating guard are marked with †. Entries are gPPL with within-sequence entropy in brackets.

Table 4: MDLM-Tiny late checkpoint, full operating-point grid. The standard (τ=1, fp64) and sharpened (τ=0.9, fp32) cells are in **bold**; † marks $H < 5$.

M	τ=0.8	τ=0.85	τ=0.9	τ=0.95	τ=1.0	τ=1.05	τ=1.10
fp64 categorical sampling.							
1	786.0† [4.55]	1276† [4.92]	2050 [5.27]	3246 [5.58]	**4961** [5.84]	7231 [6.05]	10037 [6.23]
2	428.7† [4.57]	732.8† [4.95]	1263 [5.32]	2132 [5.64]	**3496** [5.91]	5464 [6.13]	8034 [6.31]
4	199.9† [4.63]	323.9† [4.99]	561.9 [5.33]	975.9 [5.62]	**1661** [5.87]	2807 [6.10]	4607 [6.28]
8	99.0† [4.76]	154.0 [5.07]	262.9 [5.36]	440.5 [5.61]	**734.1** [5.81]	1283 [6.01]	2261 [6.19]
16	59.9† [4.88]	93.1 [5.14]	151.3 [5.37]	253.7 [5.59]	**435.1** [5.78]	748.9 [5.95]	1371 [6.13]
32	44.1† [4.91]	67.0 [5.15]	104.9 [5.36]	175.9 [5.56]	**293.3** [5.73]	508.6 [5.90]	910.6 [6.07]
64	35.9† [4.86]	55.2 [5.11]	87.5 [5.33]	144.9 [5.53]	**246.4** [5.71]	441.5 [5.87]	805.4 [6.04]
128	32.8† [4.83]	50.4 [5.09]	77.4 [5.28]	129.3 [5.50]	**230.7** [5.69]	418.8 [5.87]	747.6 [6.04]
256	32.1† [4.82]	49.6 [5.08]	77.8 [5.30]	127.9 [5.51]	**219.0** [5.68]	377.5 [5.85]	691.4 [6.03]
512	29.9† [4.80]	46.7 [5.06]	74.6 [5.29]	124.0 [5.49]	**216.6** [5.68]	384.0 [5.86]	725.4 [6.03]
1024	30.2† [4.81]	44.8 [5.04]	71.5 [5.26]	118.0 [5.47]	**207.0** [5.66]	376.4 [5.85]	693.3 [6.02]
fp32 categorical sampling.							
1	789.4† [4.55]	1288† [4.92]	**2083** [5.27]	3249 [5.58]	4922 [5.83]	7249 [6.05]	10089 [6.22]
2	422.2† [4.55]	723.3† [4.94]	**1246** [5.31]	2120 [5.63]	3451 [5.90]	5387 [6.13]	7818 [6.30]
4	193.2† [4.62]	323.3† [4.98]	**540.0** [5.32]	948.3 [5.62]	1594 [5.87]	2747 [6.09]	4504 [6.28]
8	95.6† [4.74]	150.3 [5.07]	**245.0** [5.35]	416.4 [5.59]	691.5 [5.81]	1213 [6.00]	2142 [6.19]
16	56.3† [4.84]	86.4 [5.12]	**137.7** [5.35]	221.6 [5.56]	368.7 [5.74]	654.6 [5.92]	1171 [6.09]
32	38.2† [4.82]	56.7 [5.07]	**90.7** [5.30]	148.7 [5.50]	249.6 [5.68]	433.9 [5.85]	781.8 [6.03]
64	31.5† [4.80]	45.0 [5.02]	**69.6** [5.24]	113.7 [5.44]	193.9 [5.63]	329.8 [5.80]	567.2 [5.97]
128	27.9† [4.74]	40.0† [4.99]	**59.2** [5.19]	90.3 [5.39]	148.3 [5.57]	239.7 [5.74]	414.8 [5.90]
256	24.6† [4.71]	34.5† [4.93]	**49.7** [5.12]	77.5 [5.32]	118.4 [5.49]	186.8 [5.66]	311.3 [5.83]
512	22.1† [4.63]	30.3† [4.85]	**42.9** [5.05]	61.7 [5.22]	94.7 [5.40]	147.2 [5.57]	237.5 [5.74]
1024	20.5† [4.58]	27.2† [4.78]	**36.3**† [4.94]	50.7 [5.11]	72.9 [5.29]	107.1 [5.45]	165.1 [5.62]

Table 5: MDLM-SMALL late checkpoint, full operating-point grid. The standard (τ=1, fp64) and sharpened (τ=0.9, fp32) cells are in **bold**; † marks $H < 5$.

M	τ=0.8	τ=0.85	τ=0.9	τ=0.95	τ=1.0	τ=1.05	τ=1.10
fp64 categorical sampling.							
1	759.3† [4.52]	1244† [4.90]	1992 [5.25]	3135 [5.55]	**4780** [5.82]	6957 [6.04]	9722 [6.21]
2	426.1† [4.51]	734.8† [4.92]	1280 [5.31]	2181 [5.64]	**3634** [5.93]	5780 [6.16]	8385 [6.34]
4	179.8† [4.54]	333.0† [4.96]	600.2 [5.34]	1075 [5.66]	**1931** [5.95]	3344 [6.18]	5612 [6.38]
8	86.3† [4.67]	141.0 [5.03]	236.4 [5.35]	421.5 [5.62]	**795.6** [5.88]	1561 [6.10]	2938 [6.30]
16	45.3† [4.82]	66.8 [5.12]	107.6 [5.38]	186.6 [5.59]	**333.9** [5.78]	615.9 [5.97]	1262 [6.17]
32	29.2† [4.90]	41.3 [5.14]	64.7 [5.35]	106.1 [5.54]	**194.5** [5.72]	355.8 [5.91]	761.4 [6.10]
64	23.1† [4.88]	34.4 [5.13]	52.3 [5.34]	84.0 [5.51]	**144.0** [5.69]	266.8 [5.86]	573.7 [6.04]
128	20.3† [4.88]	29.0 [5.11]	44.3 [5.31]	71.9 [5.50]	**121.4** [5.67]	225.8 [5.83]	461.7 [6.01]
256	18.8† [4.82]	27.9 [5.08]	42.4 [5.30]	70.1 [5.48]	**122.1** [5.66]	237.8 [5.84]	496.5 [6.03]
512	17.9† [4.79]	25.5 [5.05]	38.0 [5.26]	63.7 [5.46]	**111.9** [5.64]	207.7 [5.82]	425.9 [6.00]
1024	18.1† [4.84]	25.5 [5.07]	38.6 [5.26]	62.9 [5.46]	**108.2** [5.63]	189.0 [5.79]	384.8 [5.97]
fp32 categorical sampling.							
1	785.0† [4.52]	1249† [4.90]	**2045** [5.25]	3197 [5.55]	4733 [5.82]	6887 [6.04]	9636 [6.21]
2	396.9† [4.48]	704.9† [4.90]	**1219** [5.29]	2104 [5.64]	3513 [5.92]	5488 [6.16]	8080 [6.34]
4	174.3† [4.51]	312.9† [4.93]	**589.6** [5.32]	1040 [5.65]	1870 [5.94]	3351 [6.19]	5456 [6.38]
8	81.2† [4.63]	132.1 [5.00]	**231.8** [5.32]	400.8 [5.60]	725.3 [5.84]	1396 [6.07]	2633 [6.27]
16	42.7† [4.77]	66.3 [5.10]	**96.7** [5.35]	169.1 [5.57]	305.0 [5.76]	537.1 [5.94]	1166 [6.15]
32	28.0† [4.89]	40.7 [5.14]	**60.9** [5.34]	93.1 [5.51]	151.2 [5.68]	283.7 [5.85]	560.5 [6.03]
64	19.6† [4.78]	28.4 [5.04]	**43.7** [5.28]	65.7 [5.45]	105.5 [5.62]	181.0 [5.77]	365.7 [5.94]
128	16.8† [4.76]	24.0 [5.00]	**33.8** [5.20]	51.2 [5.38]	80.1 [5.55]	128.6 [5.70]	243.1 [5.87]
256	15.4† [4.70]	21.1† [4.95]	**29.3** [5.14]	41.8 [5.30]	64.3 [5.47]	107.8 [5.63]	181.5 [5.78]
512	13.7† [4.63]	18.0† [4.84]	**24.7** [5.05]	35.4 [5.24]	50.1 [5.38]	80.1 [5.54]	121.0 [5.69]
1024	12.9† [4.61]	16.7† [4.82]	**22.5** [5.03]	31.0 [5.18]	41.9 [5.33]	61.5 [5.46]	96.0 [5.60]

Table 6: MDLM-SMALL early checkpoint, full operating-point grid. The standard (τ=1, fp64) and sharpened (τ=0.9, fp32) cells are in **bold**; † marks $H < 5$.

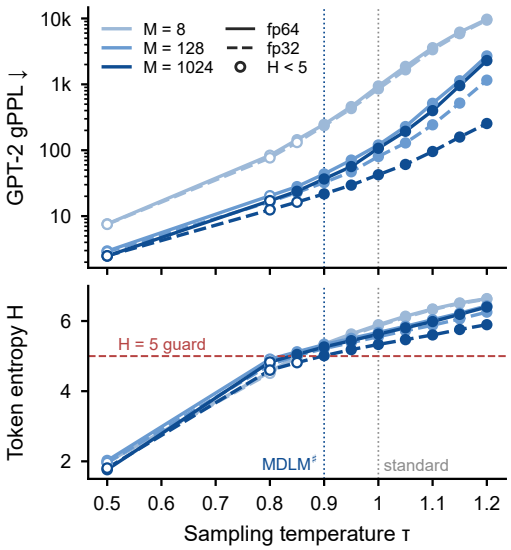
M	τ=0.8	τ=0.85	τ=0.9	τ=0.95	τ=1.0	τ=1.05	τ=1.10
fp64 categorical sampling.							
1	782.9† [4.48]	1251† [4.87]	2007 [5.22]	3162 [5.54]	**4941** [5.80]	7116 [6.03]	9854 [6.20]
2	431.3† [4.55]	703.7† [4.94]	1212 [5.31]	2022 [5.62]	**3278** [5.89]	5088 [6.11]	7647 [6.29]
4	189.0† [4.72]	305.5 [5.07]	507.2 [5.38]	835.1 [5.63]	**1376** [5.86]	2313 [6.07]	3891 [6.25]
8	96.1† [4.90]	144.2 [5.17]	235.2 [5.41]	365.8 [5.61]	**605.9** [5.80]	1024 [5.99]	1801 [6.15]
16	55.3† [4.94]	85.3 [5.19]	131.1 [5.38]	211.5 [5.57]	**343.1** [5.73]	581.9 [5.91]	1052 [6.07]
32	39.7† [4.93]	56.5 [5.14]	88.1 [5.34]	146.1 [5.53]	**251.2** [5.70]	390.4 [5.86]	732.5 [6.02]
64	33.0† [4.91]	49.7 [5.13]	75.1 [5.32]	123.8 [5.51]	**222.1** [5.69]	377.4 [5.85]	685.9 [6.02]
128	29.9† [4.90]	44.8 [5.10]	70.2 [5.30]	111.8 [5.48]	**190.0** [5.67]	325.0 [5.83]	573.6 [5.99]
256	28.5† [4.89]	43.4 [5.10]	68.8 [5.31]	110.7 [5.47]	**185.3** [5.64]	328.7 [5.83]	614.7 [6.00]
512	26.8† [4.83]	38.8 [5.05]	60.1 [5.25]	98.1 [5.44]	**180.1** [5.64]	315.1 [5.81]	614.4 [5.99]
1024	26.4† [4.84]	40.0 [5.07]	57.5 [5.23]	96.0 [5.43]	**166.1** [5.62]	292.0 [5.79]	519.3 [5.96]
fp32 categorical sampling.							
1	798.5† [4.48]	1281† [4.87]	**2085** [5.23]	3275 [5.54]	4864 [5.80]	7041 [6.03]	9793 [6.21]
2	404.8† [4.52]	686.8† [4.93]	**1151** [5.29]	1904 [5.61]	3105 [5.88]	4811 [6.11]	7258 [6.28]
4	178.3† [4.71]	297.5 [5.06]	**498.2** [5.37]	815.3 [5.63]	1385 [5.86]	2304 [6.06]	3731 [6.24]
8	91.7† [4.88]	138.6 [5.15]	**213.7** [5.39]	344.6 [5.59]	551.5 [5.78]	923.9 [5.96]	1665 [6.14]
16	52.5† [4.93]	76.2 [5.16]	**120.5** [5.37]	195.1 [5.55]	324.6 [5.72]	531.1 [5.90]	956.4 [6.05]
32	36.8† [4.92]	53.6 [5.13]	**81.7** [5.32]	131.1 [5.49]	205.1 [5.65]	349.4 [5.82]	586.1 [5.97]
64	27.9† [4.81]	42.2 [5.06]	**61.7** [5.26]	96.3 [5.43]	152.1 [5.60]	263.3 [5.76]	442.0 [5.91]
128	24.5† [4.78]	34.8† [4.99]	**50.4** [5.17]	76.7 [5.35]	120.6 [5.53]	201.7 [5.69]	345.3 [5.86]
256	22.7† [4.74]	30.6† [4.92]	**42.8** [5.11]	63.1 [5.28]	94.0 [5.45]	154.7 [5.61]	246.6 [5.76]
512	19.3† [4.63]	26.7† [4.84]	**36.2** [5.02]	50.8 [5.16]	75.3 [5.35]	112.1 [5.51]	177.5 [5.67]
1024	18.5† [4.59]	24.0† [4.79]	**32.1**† [4.95]	44.6 [5.13]	64.4 [5.30]	93.2 [5.44]	134.4 [5.58]

Table 7: **DiDi-Instruct, unguided ancestral sampling.** GPT-2-Large gPPL [H], 40 samples per cell. Bold: standard $(1, \text{fp64})$ and sharpened $(.9, \text{fp32})$; †: $H < 5$.

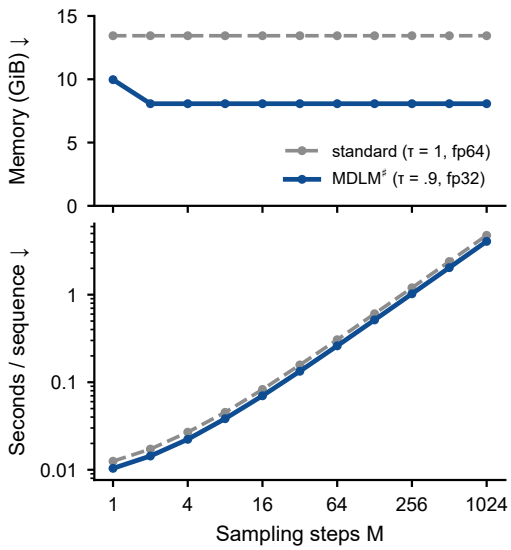
M	τ=0.8	τ=0.85	τ=0.9	τ=0.95	τ=1.0	τ=1.05	τ=1.10
fp64 categorical sampling.							
1	506.6† [4.44]	832.6† [4.79]	1393 [5.15]	2268 [5.46]	**3371** [5.72]	4958 [5.95]	7007 [6.13]
2	288.3† [4.26]	483.8† [4.70]	866.1 [5.12]	1538 [5.50]	**2487** [5.79]	3958 [6.05]	6114 [6.25]
4	110.4† [4.38]	160.8† [4.80]	241.1 [5.10]	367.1 [5.36]	**532.5** [5.56]	757.3 [5.72]	1168 [5.90]
8	46.07† [4.72]	57.75† [4.94]	77.23 [5.15]	106.9 [5.31]	**148.2** [5.46]	192.7 [5.58]	281.9 [5.72]
16	25.82† [4.87]	32.36 [5.02]	40.25 [5.17]	52.63 [5.30]	**69.50** [5.41]	91.69 [5.53]	124.4 [5.63]
32	19.01† [4.88]	22.66 [5.03]	28.14 [5.16]	36.23 [5.27]	**44.80** [5.37]	61.95 [5.52]	83.03 [5.60]
64	15.63† [4.87]	19.29 [5.05]	24.17 [5.16]	30.31 [5.26]	**37.68** [5.36]	49.26 [5.47]	69.53 [5.56]
128	14.54† [4.88]	16.47† [4.97]	20.51 [5.11]	25.56 [5.22]	**34.42** [5.36]	44.01 [5.47]	60.19 [5.58]
256	13.22† [4.74]	15.94† [4.94]	19.52 [5.04]	23.69 [5.17]	**28.99** [5.26]	40.14 [5.44]	54.84 [5.53]
512	12.18† [4.72]	14.78† [4.90]	18.35 [5.04]	23.06 [5.14]	**30.15** [5.29]	40.63 [5.42]	54.27 [5.49]
1024	11.71† [4.71]	14.73† [4.93]	19.40 [5.09]	24.15 [5.20]	**30.28** [5.30]	41.02 [5.46]	55.21 [5.55]
fp32 categorical sampling.							
1	463.8† [4.40]	761.6† [4.76]	**1287** [5.11]	2138 [5.44]	3289 [5.70]	4874 [5.92]	6941 [6.12]
2	279.3† [4.22]	457.6† [4.66]	**842.3** [5.11]	1432 [5.46]	2388 [5.76]	3875 [6.02]	5987 [6.24]
4	107.9† [4.41]	153.3† [4.79]	**235.5** [5.12]	357.1 [5.36]	510.3 [5.54]	792.1 [5.75]	1194 [5.90]
8	44.99† [4.69]	57.98† [4.95]	**74.63** [5.13]	103.0 [5.31]	139.1 [5.42]	195.2 [5.58]	287.3 [5.72]
16	26.24† [4.88]	32.68 [5.01]	**42.68** [5.17]	55.42 [5.29]	67.49 [5.40]	88.35 [5.51]	122.6 [5.63]
32	18.51† [4.87]	21.89 [5.03]	**26.75** [5.14]	35.68 [5.26]	42.60 [5.38]	58.88 [5.48]	74.58 [5.58]
64	14.54† [4.83]	18.02† [4.98]	**21.66** [5.07]	28.21 [5.22]	33.67 [5.33]	44.21 [5.43]	59.08 [5.54]
128	13.05† [4.78]	16.60† [4.97]	**19.92** [5.09]	24.31 [5.20]	31.25 [5.31]	37.76 [5.38]	50.78 [5.50]
256	12.12† [4.72]	14.17† [4.84]	**17.22**† [4.99]	20.79 [5.10]	26.15 [5.23]	33.92 [5.34]	44.05 [5.45]
512	10.97† [4.67]	13.25† [4.88]	**16.31**† [4.99]	20.39 [5.13]	24.45 [5.22]	31.73 [5.33]	39.61 [5.43]
1024	10.69† [4.57]	12.88† [4.76]	**15.23**† [4.91]	17.46 [5.00]	20.96 [5.10]	26.68 [5.22]	34.57 [5.36]

Table 8: **DiDi-Instruct, reward-guided ancestral sampling.** GPT-2-Large gPPL [H], 40 samples per cell. Bold: standard $(1, \text{fp64})$ and sharpened $(.9, \text{fp32})$; †: $H < 5$.

M	τ=0.8	τ=0.85	τ=0.9	τ=0.95	τ=1.0	τ=1.05	τ=1.10
fp64 categorical sampling.							
1	513.8† [4.46]	849.1† [4.82]	1478 [5.19]	2357 [5.50]	**3474** [5.73]	5343 [5.97]	7516 [6.16]
2	294.3† [4.28]	508.2† [4.73]	918.0 [5.17]	1607 [5.54]	**2550** [5.81]	4074 [6.07]	6249 [6.27]
4	109.9† [4.41]	154.2† [4.80]	244.4 [5.12]	368.7 [5.37]	**523.6** [5.56]	771.5 [5.74]	1261 [5.93]
8	43.88† [4.64]	61.45† [4.90]	77.23 [5.15]	103.7 [5.31]	**147.9** [5.45]	196.8 [5.56]	295.1 [5.72]
16	24.59† [4.82]	31.50† [4.99]	40.95 [5.17]	52.10 [5.29]	**71.56** [5.43]	90.89 [5.52]	119.8 [5.62]
32	20.04† [4.88]	22.88 [5.02]	27.64 [5.13]	36.92 [5.28]	**46.76** [5.39]	64.35 [5.50]	83.71 [5.59]
64	15.24† [4.88]	18.76 [5.01]	23.41 [5.15]	27.85 [5.23]	**37.28** [5.37]	50.21 [5.46]	64.89 [5.55]
128	14.20† [4.83]	16.85† [4.98]	19.73 [5.10]	25.19 [5.22]	**34.51** [5.35]	42.10 [5.43]	58.36 [5.55]
256	13.42† [4.85]	15.39† [4.92]	18.42 [5.04]	23.93 [5.21]	**30.93** [5.33]	40.56 [5.43]	54.13 [5.55]
512	11.92† [4.73]	15.25† [4.93]	18.60 [5.07]	24.22 [5.22]	**30.82** [5.33]	41.03 [5.44]	53.36 [5.53]
1024	11.84† [4.72]	14.62† [4.89]	17.71 [5.05]	23.05 [5.20]	**30.06** [5.31]	39.70 [5.43]	58.32 [5.57]
fp32 categorical sampling.							
1	497.6† [4.47]	821.1† [4.82]	**1374** [5.20]	2269 [5.51]	3438 [5.76]	5062 [5.98]	7214 [6.17]
2	300.2† [4.29]	501.9† [4.73]	**876.5** [5.16]	1526 [5.51]	2448 [5.80]	4008 [6.05]	5990 [6.26]
4	105.1† [4.44]	156.7† [4.84]	**220.1** [5.12]	335.0 [5.35]	483.0 [5.53]	756.6 [5.73]	1107 [5.89]
8	43.88† [4.75]	57.22† [4.96]	**76.09** [5.14]	102.5 [5.32]	141.4 [5.43]	195.5 [5.58]	278.8 [5.71]
16	24.93† [4.85]	31.53 [5.02]	**40.66** [5.17]	51.59 [5.28]	67.30 [5.39]	88.09 [5.52]	123.5 [5.62]
32	17.77† [4.86]	21.54† [4.99]	**26.73** [5.13]	33.99 [5.25]	42.33 [5.37]	56.25 [5.48]	76.96 [5.59]
64	14.55† [4.81]	16.91† [4.92]	**20.74** [5.07]	27.05 [5.21]	36.44 [5.35]	44.32 [5.45]	59.53 [5.56]
128	12.99† [4.77]	15.72† [4.92]	**18.91** [5.05]	24.35 [5.20]	31.04 [5.31]	38.68 [5.43]	50.67 [5.52]
256	11.91† [4.70]	14.78† [4.92]	**17.04** [5.03]	20.12 [5.13]	27.18 [5.27]	34.44 [5.37]	48.18 [5.49]
512	10.98† [4.62]	13.21† [4.81]	**15.88**† [4.97]	18.93 [5.08]	23.29 [5.20]	29.85 [5.33]	39.01 [5.43]
1024	11.66† [4.74]	13.29† [4.83]	**15.08**† [4.95]	17.49 [5.05]	21.28 [5.14]	26.19 [5.27]	33.09 [5.37]



(a) τ-sweep of the public checkpoint at $M \in \{8, 128, 1024\}$. gPPL falls under sharpening (top), while token entropy eventually crosses the $H \geq 5$ guard (bottom); filled markers pass the guard and open markers fail it. Dotted lines mark the standard $\tau = 1$ and the MDLM$^\sharp$ $\tau = .9$.



(b) Matching-configuration cost of MDLM$^\sharp$ (τ=.9, fp32) and the standard (τ=1, fp64) sampler: peak allocated memory (top) and amortized wall-clock time (bottom).

Figure 5: **The sampler exposes an empirical quality–diversity–cost frontier.** *Left:* gPPL and generated-sequence H show why neither metric alone suffices. *Right:* cost is measured on the same H100 protocol for the two displayed operating points.

A.5 The sharpened sampler is also memory and compute efficient

Figure 5b is a matching-configuration comparison of MDLM$^\sharp$ (τ=.9, fp32) against the standard (τ=1, fp64) sampler on MDLM-Small for $M = 1, \ldots, 1024$. At $M = 1024$, MDLM$^\sharp$ uses 8.069 GiB peak allocated memory versus 13.439 GiB and takes 4.065 versus 4.764 amortized wall-seconds per sequence at batch size 8: 40.0% lower peak allocation and 17.2% higher throughput. Each point uses the H100 protocol above. This run includes the extra log-softmax path invoked by $\tau \neq 1$, so the comparison exactly matches the operating points shown in the quality figures.

A.6 The unchanged baseline rivals its distilled successors

Full cross-model comparison. Table 9 gives the full comparison behind Figure 1a, in gPPL with within-sequence entropy in brackets.

Entropy guard diagnostic. Figure 6 plots within-sequence entropy versus configured M for the cross-model stress test. MDLM$^\sharp$ remains above the guard through $M = 256$ and is just below it at $M = 1024$ ($H = 4.98$); sharpened DiDi-Instruct remains above it through $M = 256$ and also crosses slightly at $M \geq 512$ ($H \approx 4.95$ at $M = 1024$). A $\tau = .95$ DiDi variant passes the guard throughout and still improves substantially over the standard sampler (Table 8). We mark the crossings rather than treating these diagnostics as a certificate of across-sample diversity or distributional correctness.

A.7 Sharpening worsens the schedule-matched posterior loss

We measure the per-position forced-mask objective $\mathcal{J}_{M+1}/L$ from (40) on 512 paired held-out sequences for the public, early, and late MDLM-Small checkpoints. We load EMA weights and use the idealized deployed reveal masses: M masses $(1 - 10^{-5})/M$ plus terminal cleanup mass 10^{-5}, hence $M + 1$ reveal blocks. Each sequence shares its corruption across temperatures; 95% intervals

Table 9: **GPT-2-Large gPPL [within-sequence entropy] in one consolidated 40-sample stress test.** M is configured reverse updates, not NFE; each official sampler uses its terminal-noise-removal convention. Sharpened rows fix $(\tau{=}.9, \text{fp32})$; all other rows use the standard $(\tau{=}1, \text{fp64})$ sampler. DUO-DCD is the distilled DUO model. †: $H < 5$. MDLM uses the public checkpoint; DiDi uses the EMA student and reward-guided sampler.

Model	$M{=}4$	$M{=}16$	$M{=}64$	$M{=}128$	$M{=}256$	$M{=}1024$
MDLM-Small	1948.8 [5.95]	344.2 [5.79]	145.4 [5.71]	117.7 [5.66]	109.9 [5.64]	103.8 [5.64]
MDLM$^\sharp$ (same checkpoint)	**519.4** [5.29]	**103.6** [5.36]	**44.5** [5.30]	**33.3** [5.23]	**30.1** [5.15]	**21.4**† [4.98]
DUO	518.3 [5.55]	126.2 [5.59]	86.2 [5.60]	86.2 [5.57]	70.1 [5.54]	85.6 [5.57]
DUO-DCD	314.9 [5.51]	83.1 [5.55]	66.0 [5.52]	62.0 [5.50]	51.2 [5.46]	65.7 [5.47]
SEDD	1863.3 [5.91]	385.7 [5.81]	146.5 [5.69]	131.9 [5.66]	100.1 [5.62]	114.0 [5.65]
DiDi-Instruct	523.6 [5.56]	71.6 [5.43]	37.3 [5.37]	34.5 [5.35]	30.9 [5.33]	30.1 [5.31]
DiDi-Instruct, sharpened	**220.1** [5.12]	**40.7** [5.17]	**20.7** [5.07]	**18.9** [5.05]	**17.0** [5.03]	**15.1**† [4.95]
GPT-2 generator		31.3 [5.50]	(update-independent reference)			

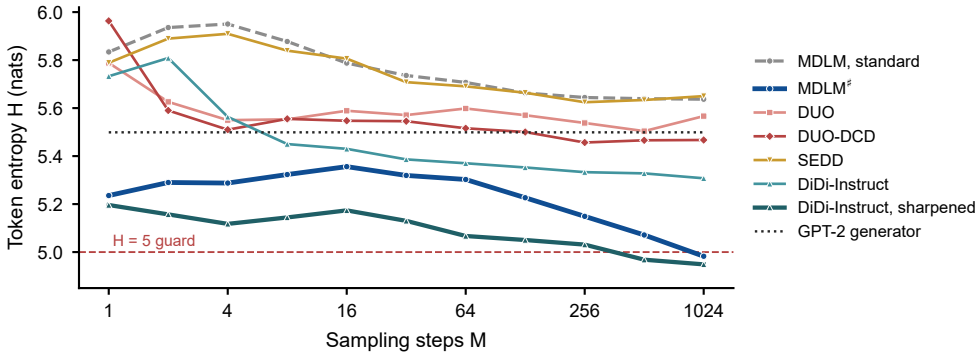


Figure 6: **Sharpened samplers stay near the $H{=}5$ guard.** Within-sequence entropy versus configured updates in the cross-model stress test. The dashed line marks $H = 5$ and the dotted line the autoregressive GPT-2 generator; MDLM$^\sharp$ and sharpened DiDi-Instruct cross the guard slightly at the largest budgets.

use a paired sequence bootstrap with 10,000 replicates. A separate 32-level, 128-sequence profile provides the early/late shape diagnostic.

Table 10: **Sharpening worsens the schedule-matched posterior loss.** Entries are $\{\mathcal{J}_{M+1}(\tau{=}.9) - \mathcal{J}_{M+1}(\tau{=}1)\}/L$ in per-position nats with paired 95% intervals. Positive values mean worse teacher-forced cross-entropy. Every interval excludes zero; the temperature-grid optimum is 1 or 1.05.

Checkpoint	$M{=}2$	$M{=}8$	$M{=}32$	$M{=}128$	$M{=}1024$
Public	$+.038$ [.011, .064]	$+.052$ [.029, .076]	$+.045$ [.021, .068]	$+.051$ [.028, .073]	$+.036$ [.014, .058]
Early	$+.047$ [.021, .075]	$+.055$ [.030, .081]	$+.041$ [.016, .068]	$+.068$ [.042, .094]	$+.037$ [.012, .061]
Late	$+.041$ [.014, .067]	$+.056$ [.033, .081]	$+.048$ [.024, .072]	$+.055$ [.031, .078]	$+.043$ [.020, .066]

At $\tau = 1$, late-minus-early differences are $-0.426, -0.546, -0.589, -0.617, -0.512$ nats at the five displayed budgets, with all paired intervals below zero. On the profile measurement, late is lower at every nontrivial clean-probability level; its left-grid mean is lower by 0.674 nats (95% CI 0.629–0.722). Thus neither the output improvement from $\tau = .9$ nor the observed early/late gPPL reversal is explained by a lower posterior loss.

A.8 LARGE DIFFUSION LMS ALREADY SIT AT THEIR SHARP LIMIT

We evaluate LLaDA-8B-Instruct (Nie et al., 2025) and Dream-v0-Base-7B (Ye et al., 2025) on 256 GSM8K (Cobbe et al., 2021) examples and all 164 HumanEval (Chen et al., 2021) problems, using the Fast-dLLM protocol (Wu et al., 2026): fp32 sampling, length 256, block size 32. We sweep $\tau \in \{0, .25, .5, .75, 1, 1.5, 2\}$, with one greedy run and three explicit seeds at positive temperatures. HumanEval outputs are sanitized into complete programs before functional evaluation; the stock continuation scorer is inappropriate for these outputs.

Greedy is best for both models on GSM8K and for Dream on HumanEval. LLaDA's HumanEval fluctuations through $\tau = 1$ are small relative to seed uncertainty; both models fail at sufficiently high temperature (Figure 7). These strong greedy defaults delimit the positive small-model finding: sharpening a positive-temperature baseline is not evidence that instruction-tuned defaults need changing.

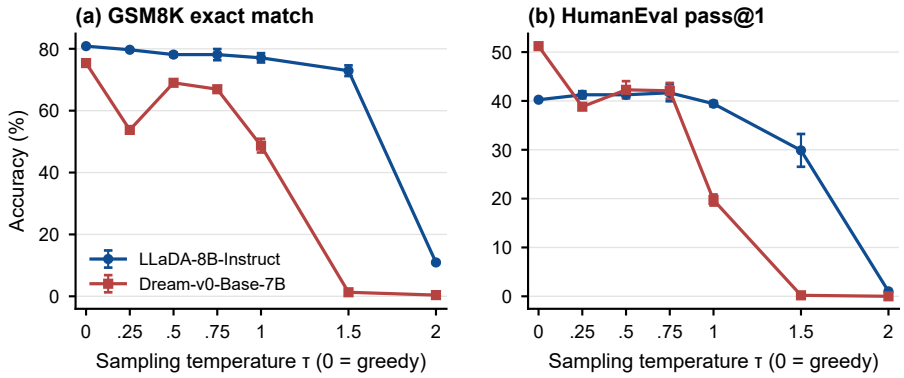


Figure 7: **For large diffusion LMs, sharpened sampler with lower temperature improves downstream performance.** LLaDA-8B-Instruct and Dream-v0-Base-7B on (a) GSM8K and (b) HumanEval. Greedy (temperature=0) is the default setting in official codebases.

Autoregressive control. Four autoregressive checkpoints use the same task, length cap, and seed protocol, with unrestricted categorical sampling above $\tau = 0$ and thinking disabled for Qwen and Gemma (Figure 8). Their moderate-temperature responses also vary by model and task; high-temperature failure is not peculiar to diffusion. The diffusion models' greedy HumanEval scores $(40.2\%, 51.2\%)$ lie within the autoregressive range $(23.8\%–56.7\%)$, but different checkpoints and post-training preclude a controlled model ranking.

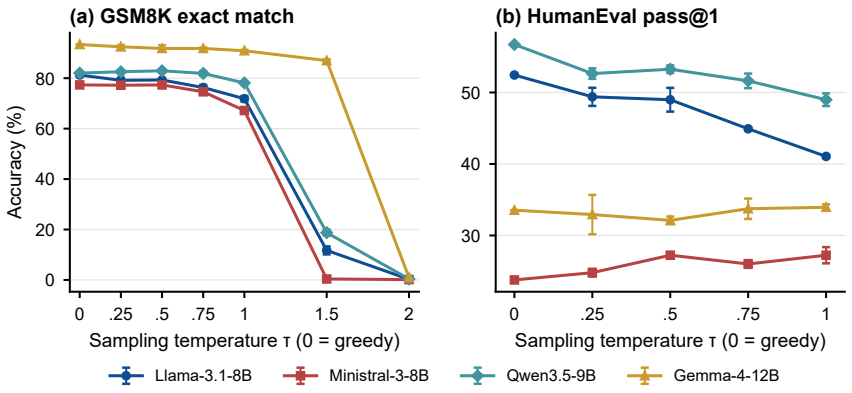


Figure 8: **Autoregressive control.** Llama-3.1-8B-Instruct, Ministral-3-8B-Instruct, Qwen3.5-9B, and Gemma-4-12B-it on (a) GSM8K, through $\tau = 2$, and (b) HumanEval, through $\tau = 1$, after complete-program extraction. Greedy runs once, while stochastic markers show three-seed means ± 1 seed SEM.

B GroupEval: Protocol, Full Results, and Diagnostics

This appendix gives the proofs and protocol behind GroupEval, followed by the full judge-separated results. Sampler names follow Section A.1, with every contrast comparing the standard $(1, \mathrm{fp64})$ with the sharpened $(.9, \mathrm{fp32})$ setting.

B.1 Proof of Proposition 1

Let the output space $\mathcal{X}$ be finite and write $z(x) = (f(x), g(x)) \in \mathbb{R}^2$. The map $Q \mapsto (\mathbb{E}_Q f, \mathbb{E}_Q g)$ is linear on the simplex $\Delta(\mathcal{X})$, so its image is the convex polygon $Z = \mathrm{conv}\{z(x) : x \in \mathcal{X}\}$. A Pareto-optimal value cannot be an interior point of Z, since a small move inside Z would improve both coordinates. Every boundary point of a polygon lies on an edge $[z(x), z(x')]$ and is attained by a generator supported on $\{x, x'\}$.

For the constrained problem, $\min \mathbb{E}_Q f$ subject to $\mathbb{E}_Q g \geq h$, $\sum_x Q(x) = 1$, and $Q \geq 0$ is a feasible, bounded linear program with two constraints besides nonnegativity whenever $h \leq \max_x g(x)$; it therefore has an optimal basic solution with at most two nonzero coordinates. Token-pooled log-gPPL on fixed-length outputs and mean per-output unigram entropy are both of this form.

Finally, if a generator places all its mass on two outputs, every group of $n = 8$ samples contains at most two distinct texts. Identical texts share a mode under the rubric, so the group's mode entropy is at most $\log 2$ and $\mathrm{GED}_g \leq 10 \log 2 / \log 8 = 10/3$.

B.2 Elementary properties and limits of GED

For a fixed group size $n \geq 2$, let valid-mode counts be $n_1, \ldots, n_K$ with $V = \sum_k n_k \leq n$. Multiplying entropy by the valid count rewrites the group score as

$$\mathrm{GED}_g = \frac{10}{n \log n} \left[V \log V - \sum_k n_k \log n_k \right], \tag{8}$$

with the continuous convention $0 \log 0 = 0$ and zero for an empty group. This expression also gives zero for a single valid sample or a single mode.

Range, invariances, and mode merging. The score is invariant to sample order and cluster labels. Since mode entropy lies between zero and $\log V$, $0 \leq \mathrm{GED}_g \leq 10(V/n)(\log V / \log n) \leq 10$ for $V \geq 2$. Equality at ten holds exactly when $V = n$ and all n modes are singletons. Merging counts $x, y > 0$ replaces $x \log x + y \log y$ by $(x + y) \log(x + y)$, a larger quantity, so it cannot increase GED.

The validity gate does not reward removal at fixed assignments. Invalidating one sample in a mode of size $m \leq V$ changes the bracket in (8) by subtracting

$$f(V) - f(m), \qquad f(x) = x \log x - (x - 1) \log(x - 1). \tag{9}$$

For $x > 1$, $f'(x) = \log(x/(x-1)) > 0$, and the endpoint is interpreted continuously. Thus $f(V) \geq f(m)$ and the score cannot increase. This proves the stated monotonicity under fixed assignments. A judge may recluster the remaining outputs when a group changes, so the property does not assert empirical robustness of the clustering procedure.

Local saturation is not support coverage. Suppose all outputs are valid and their true modes are independently uniform on $K \geq n$ modes. The probability that all group modes are distinct is $\prod_{j=0}^{n-1}(1 - j/K)$, at least $1 - \binom{n}{2}/K$ by a collision union bound. On that event GED is ten. As K grows, a fixed small group therefore almost always saturates, while observing a vanishing fraction of the total modes. This is why GroupEval reports ceiling rates and does not interpret high GED as recovery of the data distribution.

Why weighted GED ranges are not ordinary confidence intervals. Consider an all-valid, all-singleton group. The unweighted GED is ten. If its sample weights are replaced by a nonuniform Dirichlet draw W and normalized within the group, its entropy becomes $H(W) < \log n$ almost

surely. The resulting weighted score is strictly below ten even though the labels have not changed. A percentile range from those draws can therefore exclude the observed score systematically. We retain such ranges only as conditional weight sensitivity summaries in the released aggregates, not as calibrated intervals around plug-in GED. GEQ, being an average of fixed quality scores rather than an entropy functional, does not have this particular centering issue.

B.3 Evaluation protocol

B.3.1 Rubric and deterministic scores

GroupEval v1.0 evaluates individual quality and local across-output diversity separately. The frozen prompt requests structured rubric decisions rather than a direct overall preference or an explanation of which system should win.

Quality. For sample i in grouping context r, a catastrophic-invalidity gate marks empty, predominantly corrupted, catastrophically unreadable, or non-communicative text invalid. Weak but usable text remains valid. The four dimensions are readability/local fluency, coherence/continuity, substance/target naturalness, and non-degeneration/document structure. Each receives $s_{ird} \in \{0, 1, 2\}$, with all four set to zero for an invalid output. Thus

$$q_{ir} = \frac{5}{4} \sum_{d=1}^{4} s_{ird}, \qquad \text{GEQ} = \frac{1}{S} \sum_{i=1}^{S} \frac{1}{R} \sum_{r=1}^{R} q_{ir}. \tag{10}$$

Invalid outputs lower the mean rather than disappearing through filtering. The report also includes invalidity, severe-degeneration rates, dimension means, the 10th percentile, and bottom-quartile quality.

Diversity. Every valid sample receives one hard, group-local semantic–discourse mode. Modes reflect substantive content, purpose or genre, narrative setup, argument or perspective, or discourse structure. Paraphrases, entity substitutions within one template, formatting differences, and random corruption do not create useful modes. GED is computed by (2); equivalently, (8) expresses it directly in mode counts. The score is zero for fewer than two valid outputs or fewer than two modes. The validity gate is binary: a low-quality but valid output retains full mode mass, so GEQ must accompany GED. We report effective mode count, dominant-mode share, hybrid-assignment rate, clustering confidence, all-singleton rate, and GED-ceiling rate. The last two differ when some outputs are invalid. Section B.2 proves the score's range and elementary invariances and explains local saturation.

Comparison and reuse. A common protocol makes comparisons interpretable across checkpoints, samplers, and budgets. Its identity includes the target-domain profile, generation conditions, judge version or available provenance, rubric, group size, preprocessing, grouping design, and scoring procedure. Each unconditional group contains outputs from one fixed generator setting. Validity and mode labels are inferred anew within each group; a mode label has no identity across groups. Comparisons keep these choices fixed and use the same output count where possible. Changing group size changes the collision probability and the $\log n$ normalization, so GED values at different group sizes are not interchangeable. The same judge name and group size alone do not make scores interchangeable, especially when a service alias can change. Judge series remain separate, and task-specific GEQ rubrics define different scales. The released prompts, schemas, grouping code, and deterministic scorer support reuse without imposing a universal quality–diversity threshold.

B.3.2 Outputs, judges, and uncertainty

The shared unconditional configuration. Every main-text unconditional GroupEval number uses the following configuration, which we recommend as a common profile for unconditional text generators. This configuration takes into account practical limitations, especially compute costs.

- **Groups.** $n = 8$ outputs from one generator setting, randomly ordered. Each output pool is partitioned twice; the second partition is chosen from 2,000 deterministic candidates to minimize reuse of output pairs. Every output is judged twice, and none is filtered.

- **Judge.** Open-weight `gpt-oss-120b` served with vLLM, with tools and web access disabled; the serving fingerprint is recorded. Generator identity, sampler, budget, and automatic scores are hidden, and candidate text is delimited as untrusted data.
- **Rubric.** The frozen v1.0 web-text profile: an invalidity gate, four quality dimensions on $\{0, 1, 2\}$, and one hard semantic–discourse mode per valid output, returned in a validated JSON schema.
- **Scores.** GEQ ((10)) over all outputs; GED ((2)) averaged over groups; never combined.
- **Uncertainty and diagnostics.** GEQ intervals from a 10,000-replicate partition-blocked Bayesian bootstrap; GED reported with invalidity, singleton, and ceiling rates.

Comparisons keep all of these fixed. Robustness judges (`gemma4:26b-a4b`, `qwen3:32b`, `gpt-5.6-luna`, `gpt-5.6-terra`) receive identical groups and are reported separately, never pooled.

Multiple judges across MDLM and DiDi. The MDLM study began with three open-weight judges (GPT-OSS, Gemma, Qwen); frontier Luna and Terra were then added for both MDLM and DiDi. Thus GPT-OSS, Luna, and Terra form the shared panel, with Gemma and Qwen providing additional MDLM sensitivity checks. Across the initial study's ten sampler–budget cells, pairwise Spearman correlations are .90–.95 for GEQ but .48–.90 for GED: quality rankings agree more closely than diversity rankings. These correlations include variation across budgets; the within-budget contrasts below test the sharpening gain directly.

Frozen outputs and grouping. Public MDLM-Small uses all 128 archived outputs in each of ten cells: two fixed settings at $M \in \{4, 16, 64, 256, 1024\}$. Groups of eight follow the two frozen partitions described above. Each judge evaluates 32 groups per cell, or 320 accepted judgments. The five MDLM judge series therefore contain 1,600 accepted judgments of the same output pools. Display orders are randomized independently. The archived runs record a global Lightning seed, not per-sample seeds; shared grouping indices do not establish sample-level common random numbers.

Each saved sequence has 1,024 nominal positions, including one nominal BOS. Preprocessing removes exactly one leading `<|endoftext|>` sentinel when present, preserves a failure to generate it, and renders every subsequent sentinel as `[DOCUMENT BOUNDARY]`. Nothing after a boundary is discarded. A cross-boundary topic change is not itself a coherence error; each document is assessed normally. The unconditional sweep's gPPL instead masks content after the first non-leading EOS. These columns are therefore not span-identical to GroupEval. The matched rescore below explicitly scores the full stream and retains first-document scoring as a sensitivity analysis.

Judge configuration and provenance. The dated MDLM series use `gpt-oss:120b` (July 25, 2026), `gemma4:26b-a4b` (July 26), `qwen3:32b` (July 27), and `gpt-5.6-luna` and `gpt-5.6-terra` (July 31). Each alias is a separate judge series, with no pooling or scale alignment.

Every accepted judgment passes local validation against the frozen schema, regardless of the provider's structured-output support. The archive records model metadata, timestamps, retries, and prompt/schema/profile and response hashes. The supplementary release contains aggregate scores, diagnostics, manifests, prompts, and analysis code, but not raw generations or individual judgments; exact reaggregation therefore requires the original archives. Mutable service aliases do not identify an immutable model snapshot.

Uncertainty. Unconditional contrasts use a 10,000-replicate partition-blocked Bayesian bootstrap (Rubin, 1981). Partition weights are shared across compared settings; sample weights are independent because the outputs are not sample-paired. Scores are recomputed within every weighted replicate. GEQ intervals condition on accepted judgments and the two frozen partitions: they do not include rejudging, new partitions, service drift, or new generation runs. For GED, unequal sample weights can systematically reduce entropy in singleton groups. Its weighted percentile ranges are therefore weight-sensitivity diagnostics, not confidence intervals centered on the plug-in GED. No diversity conclusion uses their exclusion of zero. The task-conditioned study uses a condition-cluster bootstrap instead.

B.4 UNCONDITIONAL GENERATION

B.4.1 MDLM-SMALL: FULL JUDGE-SEPARATED RESULTS

Table 11 brings all five judges into one comparison. At $M \geq 16$, all twenty quality contrasts favor MDLM$^\sharp$, and thirteen conditional 95% intervals exclude zero. Diversity rises in sixteen of twenty contrasts, despite lower unigram entropy in every pair. Differences use unrounded aggregates; the released CSVs retain all score and uncertainty diagnostics.

Table 11: **MDLM improves in quality under all five judges at every primary budget.** GEQ and GED are standard → MDLM$^\sharp$ on the same 128 outputs per cell. ΔGEQ is sharpened minus standard, with conditional 95% intervals; bold marks intervals excluding zero before rounding. All budgets, including the $M = 4$ floor case, are retained.

Steps M	GEQ ↑	ΔGEQ [95% interval]	GED ↑
GPT-OSS (default)			
4	$1.66 \to 1.53$	$-0.12\ [-0.61, 0.35]$	$0.79 \to 1.11$
16	$3.85 \to 4.36$	$+0.50\ [-0.08, 1.08]$	$6.00 \to 7.57$
64	$4.69 \to 5.00$	$+0.31\ [-0.21, 0.82]$	$8.28 \to 8.79$
256	$4.92 \to 5.99$	$\mathbf{+1.07\ [0.42, 1.73]}$	$8.67 \to 9.02$
1024	$5.15 \to 5.47$	$+0.32\ [-0.20, 0.85]$	$8.07 \to 8.75$
Gemma			
4	$2.93 \to 2.57$	$-0.36\ [-0.76, 0.05]$	$6.76 \to 4.52$
16	$4.94 \to 5.20$	$\mathbf{+0.26\ [0.00, 0.55]}$	$8.90 \to 9.24$
64	$5.77 \to 6.11$	$\mathbf{+0.34\ [0.04, 0.65]}$	$9.86 \to 8.93$
256	$6.22 \to 6.42$	$+0.21\ [-0.14, 0.57]$	$9.53 \to 9.14$
1024	$6.13 \to 6.43$	$+0.29\ [-0.09, 0.66]$	$9.51 \to 9.27$
Qwen			
4	$3.15 \to 4.72$	$\mathbf{+1.57\ [0.87, 2.27]}$	$2.76 \to 4.82$
16	$5.24 \to 5.28$	$+0.04\ [-0.33, 0.39]$	$7.28 \to 8.28$
64	$5.12 \to 5.85$	$\mathbf{+0.74\ [0.36, 1.11]}$	$9.21 \to 9.70$
256	$5.47 \to 6.92$	$\mathbf{+1.45\ [0.58, 2.29]}$	$9.12 \to 9.58$
1024	$5.77 \to 7.68$	$\mathbf{+1.90\ [1.33, 2.45]}$	$9.53 \to 9.80$
Luna			
4	$1.52 \to 1.92$	$+0.40\ [-0.43, 1.19]$	$1.64 \to 3.39$
16	$4.47 \to 5.37$	$\mathbf{+0.90\ [0.34, 1.49]}$	$8.77 \to 9.70$
64	$5.47 \to 5.94$	$\mathbf{+0.47\ [0.24, 0.71]}$	$9.86 \to 9.67$
256	$5.61 \to 6.06$	$\mathbf{+0.45\ [0.20, 0.71]}$	$9.59 \to 9.69$
1024	$5.69 \to 6.22$	$\mathbf{+0.53\ [0.30, 0.79]}$	$9.53 \to 9.70$
Terra			
4	$0.00 \to 0.00$	$+0.00\ [0.00, 0.00]$	$0.00 \to 0.00$
16	$1.00 \to 3.44$	$\mathbf{+2.45\ [1.55, 3.30]}$	$1.46 \to 5.60$
64	$4.72 \to 5.62$	$\mathbf{+0.90\ [0.56, 1.28]}$	$8.03 \to 8.89$
256	$5.20 \to 5.74$	$\mathbf{+0.54\ [0.24, 0.87]}$	$8.54 \to 9.61$
1024	$5.37 \to 5.65$	$+0.28\ [-0.07, 0.68]$	$8.80 \to 9.54$

Failure cases and saturation. At $M = 4$, GPT-OSS and Gemma favor the standard sampler in point estimate, Qwen and Luna favor MDLM$^\sharp$, and Terra rejects both. GPT-OSS marks 59.4% of standard and 54.7% of MDLM$^\sharp$ sample contexts invalid at this budget. Lower gPPL alone does not rescue this regime.

Under GPT-OSS, mean effective-mode counts span 1.08–6.96, all-singleton rates 9.4%–62.5%, and GED-ceiling rates 0%–50.0% across all ten cells. At $M \geq 16$, Luna's all-singleton and ceiling rates are substantially higher: 68.8%–87.5% and 65.6%–87.5%, respectively; hybrid assignments reach 84.3% and non-high confidence 87.5%. Terra's corresponding singleton and ceiling ranges are 18.8%–75.0% and 6.3%–68.8%. These diagnostics limit the resolution of small GED differences, particularly under Luna. Every cell's failure, mode, and confidence diagnostics are retained in the supplementary CSVs.

B.4.2 DiDi-Instruct: a judge-sensitive boundary case

The reward-guided DiDi-Instruct EMA student (Zheng et al., 2026) uses the same target profile, prompt, and scoring rules, but 40 outputs per cell. Two partitions into groups of eight yield ten judgments per cell, or 100 per judge. GPT-OSS, Luna, and Terra each score all ten cells. These are repeated judgments of fixed outputs, not independent generation replications. The standalone DiDi gPPL uses GPT-2-Large; it is not numerically comparable with the GPT-2 columns for the 128-output MDLM study.

Table 12: **DiDi-Instruct's perplexity gain does not yield a judge-robust quality gain.** Same layout as Table 11, with 40 outputs per cell. GPT-OSS and Terra have no resolved quality gain; Luna favors the standard sampler at two primary budgets.

Steps M	GEQ ↑	ΔGEQ [95% interval]	GED ↑
GPT-OSS (default)			
4	$1.59 \to 1.75$	$+0.16\ [-0.68, 1.00]$	$1.81 \to 1.73$
16	$3.53 \to 3.72$	$+0.19\ [-0.65, 0.96]$	$2.45 \to 4.20$
64	$4.48 \to 4.48$	$+0.00\ [-0.75, 0.70]$	$4.86 \to 7.46$
256	$4.33 \to 5.03$	$+0.70\ [-0.21, 1.57]$	$5.25 \to 7.63$
1024	$4.38 \to 5.08$	$+0.70\ [-0.16, 1.63]$	$6.04 \to 7.51$
Luna			
4	$2.59 \to 2.62$	$+0.03\ [-0.86, 0.79]$	$9.16 \to 9.82$
16	$4.86 \to 4.53$	$\mathbf{-0.33}\ [\mathbf{-0.68}, \mathbf{-0.02}]$	$8.43 \to 8.27$
64	$4.81 \to 4.50$	$-0.31\ [-0.67, 0.01]$	$9.00 \to 8.15$
256	$4.86 \to 4.84$	$-0.02\ [-0.29, 0.23]$	$9.19 \to 9.72$
1024	$4.98 \to 4.69$	$\mathbf{-0.30}\ [\mathbf{-0.59}, \mathbf{-0.03}]$	$9.72 \to 8.93$
Terra			
4	$0.00 \to 0.00$	$+0.00\ [0.00, 0.00]$	$0.00 \to 0.00$
16	$2.53 \to 3.12$	$+0.59\ [-0.10, 1.26]$	$1.62 \to 0.92$
64	$4.45 \to 4.81$	$+0.36\ [-0.24, 0.99]$	$4.69 \to 6.66$
256	$4.50 \to 4.50$	$+0.00\ [-0.49, 0.52]$	$5.14 \to 7.37$
1024	$4.58 \to 4.48$	$-0.09\ [-0.57, 0.42]$	$7.59 \to 8.89$

GPT-OSS gives the sharpened sampler a higher or tied GEQ point estimate at every budget, but all its intervals include zero. Luna instead favors the standard sampler at all four primary budgets, with intervals below zero at $M \in \{16, 1024\}$. Terra is mixed and all its intervals include zero. The DiDi quality improvement is therefore not judge-robust. GED directions are also mixed. Luna marks 76.3%–100% of valid DiDi assignments hybrid; 50%–90% of groups reach the GED ceiling and low clustering confidence reaches 90%. Its near-zero invalidity even at $M = 4$ contrasts sharply with GPT-OSS and Terra. These observations expose rubric and mode-assignment sensitivity rather than justify selecting a preferred judge.

B.4.3 Exact-output cross-generator comparison

The cross-generator study evaluates the exact 40 outputs behind each MDLM and DiDi cell in Table 9: two generators, two settings, and five budgets, totaling 800 outputs. A single globally interleaved, blinded GPT-OSS series provides 200 homogeneous-group judgments. Contrasts are matched by setting and budget and blocked by partition, not by sample index.

Both GPT-2-Large and pretrained Qwen3-8B-Base (Yang et al., 2025) rescore these same outputs with pinned evaluator revisions. The primary full-stream policy splits at every source document boundary, resets causal context, and pools token negative log-likelihoods and counts before exponentiation. First-document scoring, equal-output macro-gPPL, and bits per byte are sensitivity analyses.

At $M \geq 16$, both likelihood evaluators favor DiDi in all eight comparisons, with closely agreeing cell rankings across the 16 generator-setting-budget cells (Spearman $\rho = .982$). MDLM's GEQ is strictly higher in seven comparisons and tied in one; the intervals exclude zero for sharpened $M = 16$ and standard $M = 256$. Its GED is higher in all eight. First-document scoring retains only 42.97% of source bytes but preserves every primary token-micro and bits-per-byte cross-model sign. Aggregation still matters: under equal-output macro-gPPL on first documents, Qwen favors

Table 13: **Likelihood and judged quality disagree on the same outputs.** gPPL columns are MDLM / DiDi ratios, with ratio>1 favoring DiDi. GEQ/GED entries are MDLM / DiDi, and ΔGEQ is MDLM minus DiDi, with positive values favoring MDLM.

M	Sampler	GPT-2-Large ratio	Qwen ratio	GEQ↑	ΔGEQ [95% interval]	GED↑
16	Std.	4.58	4.69	3.77 / 3.77	0.00 [-1.03, 1.02]	6.88 / 4.27
16	Sharp.	2.33	2.71	4.34 / 3.16	1.19 [0.41, 1.99]	6.61 / 1.96
64	Std.	3.78	3.68	4.30 / 4.09	0.20 [-0.85, 1.21]	8.02 / 5.27
64	Sharp.	2.19	2.42	4.64 / 4.23	0.41 [-0.26, 1.12]	9.24 / 7.45
256	Std.	3.49	3.58	5.70 / 4.20	1.50 [0.39, 2.65]	8.70 / 6.59
256	Sharp.	1.76	1.95	5.41 / 4.69	0.72 [-0.21, 1.64]	9.27 / 6.23
1024	Std.	3.31	3.60	5.09 / 4.33	0.77 [-0.25, 1.73]	7.85 / 5.24
1024	Sharp.	1.48	1.56	5.53 / 4.47	1.06 [-0.07, 2.22]	9.30 / 7.08

MDLM in three of eight comparisons. Thus the stable primary conclusion is specific to the declared token-pooled metric, and may not extend to every aggregation rule.

Within each generator, likelihood is more informative about sampler selection. Excluding $M = 4$, descriptive GEQ/log-gPPL Spearman correlations are $-.57$ and $-.74$ for MDLM under GPT-2-Large and Qwen, respectively, and $-.93$ for DiDi under either evaluator. Pooling generators weakens these to $-.20$ and $-.18$. This demonstrates a cross-generator disagreement between constructs, not that either scale is ground truth.

B.5 Task-conditioned criterion check

We evaluate the greedy LLaDA-8B-Instruct and Dream-v0-Base-7B generations in Section A.8, using all 256 aligned GSM8K conditions and 164 aligned, sanitized HumanEval conditions. For each condition, GPT-OSS receives two randomized, model-blinded responses and scores readability, reasoning coherence, apparent task success, and non-degeneration. It sees no automatic labels, GSM8K answers, HumanEval hidden tests or canonical solutions, or aggregate results, and has no tools.

Table 14: **Conditional GEQ tracks task success and the task-dependent model ordering.** L / D: LLaDA / Dream. Intervals: 10,000-replicate condition-cluster bootstrap, conditional on one GPT-OSS judgment per response pair. AUROC pools both models; the last column uses conditions with exactly one passing response (counts in parentheses). Task-specific cGEQ is not comparable with unconditional GEQ; GED is undefined with one response per model and condition.

Task	Automatic success L / D	cGEQ L / D	ΔcGEQ D–L [95% CI]	AUROC [95% CI]	cGEQ(pass)–cGEQ(fail) [95% CI]
GSM8K	80.9 / 75.4%	9.54 / 8.99	-0.55 [-0.77, -0.34]	0.91 [0.87, 0.95]	3.04 [2.54, 3.53] (56)
HumanEval	40.2 / 51.2%	8.13 / 8.40	0.27 [-0.13, 0.65]	0.96 [0.94, 0.98]	3.07 [2.53, 3.66] (42)

cGEQ matches the automatic-success ordering on both tasks, although the aggregate HumanEval difference remains statistically unresolved. Pooled AUROC is .91 on GSM8K and .96 on HumanEval; all four task-by-model AUROCs exceed .85. On the 56 GSM8K and 42 HumanEval discordant conditions, the passing response scores about three points higher, with both intervals above zero. Apparent task success is intentionally a rubric dimension, so this is criterion-related evidence, not an independent human-preference validation. Functional execution remains authoritative for HumanEval.

Interpretation and scope. GroupEval separates judge-specific quality from local mode diversity. Multiple judges test sensitivity, not human alignment. Small groups can miss modes, and repeated judging does not replace independent generation. Human calibration of quality and validation of mode assignments remain open.

C Finite-Step Analysis

The proofs follow Section 4: Theorem 1 in Section C.2, Theorem 2 in Section C.3, and Proposition 2 in Section C.4. Supporting bounds accompany the result they explain. Section C.5 connects the theory to the posterior-loss measurement and implemented sampler.

C.1 Preliminaries: assumptions and reveal paths

Standing assumptions. All logarithms are natural. Entropy and KL use $0 \log 0 = 0$; conditional distributions on P-null events may be assigned arbitrarily. The vocabulary is finite, the sequence length L is fixed unless stated otherwise, and denoisers assign positive probability to every clean token. Exact categorical arithmetic is assumed throughout the proofs; implementation perturbations are treated separately in Section C.5.2. The denoiser in Section 4 is written $q^i(\cdot \mid x_A)$; when a learned denoiser also receives the visible probability a, every statement below holds with $q^i(\cdot \mid x_A, a)$ in its place.

Reveal rounds and the augmented law. Let $0 = a_0 < a_1 < \cdots < a_N = 1$ and $\delta_k = a_k - a_{k-1}$. The ideal ancestral kernel reveals a still-masked coordinate in round k with probability

$$\rho_k = \frac{a_k - a_{k-1}}{1 - a_{k-1}}. \tag{11}$$

Its survival probability through round k is $\prod_{j=1}^{k}(1-\rho_j) = 1 - a_k$, by telescoping. Hence the unconditional probability of first reveal in round k is δ_k. Independent reveal decisions across coordinates give independent $K_i \sim \mathrm{Cat}(\delta)$. The last round has $\rho_N = 1$, so every coordinate is ultimately revealed.

For an assignment $K = (K_1, \ldots, K_L)$, write $S_k = \{i : K_i = k\}$ and $A_{k-1} = \cup_{j<k} S_j$. Empty blocks are allowed. Let $\nu(S) = \prod_i \delta_{K_i}$. Conditional on the complete partition, the sampler's clean-sequence law is

$$Q_\lambda^S(x) = \prod_{k=1}^{N} \prod_{i \in S_k} q_\lambda^i(x_i \mid x_{A_{k-1}}, a_{k-1}), \qquad Q_\lambda(x, S) = \nu(S) Q_\lambda^S(x). \tag{12}$$

Every factor is a normalized conditional distribution; summing in reverse block order shows $\sum_x Q_\lambda^S(x) = 1$. The output law is $\pi_\lambda(x) = \sum_S \nu(S) Q_\lambda^S(x)$. Under the comparison law $P \otimes \nu$, the clean sequence and reveal partition are independent. Under the sampler they generally are not independent.

C.2 Proof of Theorem 1: training integrates, sampling sums

We first establish the two area identities, then the finite-step decomposition. The final part derives the supporting bounds used in Section 4.1.

C.2.1 The entropy area identity and the training objective

Revealing one more coordinate adds its conditional entropy. Averaging this observation over random visible sets gives the entropy profile as a derivative.

For any set function f on subsets of $[L]$, let U_a contain each coordinate independently with probability a. Its multilinear extension along the diagonal satisfies

$$\frac{d}{da}\mathbb{E} f(U_a) = \sum_{i=1}^{L} \mathbb{E}_{A \subseteq [L] \setminus \{i\}} \big[f(A \cup \{i\}) - f(A) \big], \tag{13}$$

where each element of A is retained independently with probability a. To prove this, first allow distinct probabilities $a_1, \ldots, a_L$. Conditioning on inclusion of i makes the expectation affine in a_i, with partial derivative equal to the bracketed difference. The chain rule along $a_1 = \cdots = a_L = a$ gives (13). This is an identity of finite polynomials, including at the endpoints.

Set $f(A) = H(X_A)$ and $F(a) = \mathbb{E}H(X_{U_a})$. Then

$$F'(a) = \sum_i \mathbb{E}_A[H(X_{A\cup\{i\}}) - H(X_A)] = \sum_i \mathbb{E}_A H(X_i \mid X_A) = h(a). \tag{14}$$

Since $F(0) = 0$ and $F(1) = H(X)$, integration proves $\int_0^1 h(a)\,da = H(X)$. The degree of h is at most $L - 1$.

At each visible context, cross-entropy splits into entropy and KL:

$$e_\lambda(a) = \sum_i \mathbb{E}_{A,X_A}\,\mathrm{KL}\big(P(X_i \in \cdot \mid X_A) \,\big\|\, q^i_\lambda(\cdot \mid X_A, a)\big) \geq 0. \tag{15}$$

This proves $c_\lambda = h + e_\lambda$ without an assumption that the denoiser is Bayes-optimal.

For the continuous-time connection, let the ideal masking schedule α_t decrease absolutely continuously from $\alpha_0 = 1$ to $\alpha_1 = 0$, and use the denoiser at the corresponding visible probability $a = \alpha_t$. At an interior time, the probability that a chosen coordinate is masked is $1 - \alpha_t$. Conditional on its masking, the other coordinates remain independently visible with probability α_t. Therefore the endpoint-completed negative ELBO functional is

$$\begin{aligned}\mathcal{L}_\infty(\lambda) &= \mathbb{E}_X \int_0^1 \frac{-\alpha_t'}{1-\alpha_t}\mathbb{E}_{Z_t|X} \sum_i \mathbf{1}\{Z_t^i = m\}[-\log q^i_\lambda(X_i \mid Z_t, \alpha_t)]\,dt \\ &= \int_0^1 (-\alpha_t')c_\lambda(\alpha_t)\,dt = \int_0^1 c_\lambda(a)\,da. \end{aligned} \tag{16}$$

The cancellation occurs before evaluating endpoint values. Measurability and integrability of c_λ suffice; if the integral is infinite, the nonnegative identity still holds in the extended sense. In particular, $\mathcal{L}_\infty(\lambda) - H(X) = \int_0^1 e_\lambda(a)\,da$. This population functional does not assert that a clipped implementation logs the omitted endpoints.

C.2.2 The finite-step decomposition

Step 1: score a fixed reveal path. From (12), cancellation of ν yields

$$\mathrm{KL}(P \otimes \nu \,\|\, Q_\lambda) = \mathbb{E}_{S,X}\left[\log P(X) - \sum_k \sum_{i\in S_k} \log q^i_\lambda(X_i \mid X_{A_{k-1}}, a_{k-1})\right]. \tag{17}$$

Conditioned on $K_i = k$, the events $\{j \in A_{k-1}\}$ for $j \neq i$ are independent with probability $\sum_{r<k}\delta_r = a_{k-1}$. The conditional distribution of A_{k-1} is consequently exactly the forced-mask context law in (3). Since $\Pr(K_i = k) = \delta_k$, the expected cross-entropy sum in (17) equals $\sum_k \delta_k c_\lambda(a_{k-1}) = \mathcal{J}_N(\lambda)$. Also $\mathbb{E}\log P(X) = -H(X)$. This proves (4) and, by $c_\lambda = h + e_\lambda$ and the area identity, its decomposition into $B_N + C_N(\lambda)$.

Step 2: identify the cost of same-round independence. To identify the parallelization term directly, define conditional total correlation by

$$\mathrm{TC}(X_B \mid X_A) = \sum_{i\in B} H(X_i \mid X_A) - H(X_B \mid X_A). \tag{18}$$

It is the expected KL between the joint conditional law and the product of its marginals, hence is nonnegative. The empty-block value is zero. For every ordered partition, $\sum_k H(X_{S_k} \mid X_{A_{k-1}}) = H(X)$ by the entropy chain rule. Thus

$$\begin{aligned}\mathbb{E}_S \sum_k \mathrm{TC}(X_{S_k} \mid X_{A_{k-1}}) &= \mathbb{E}_S \sum_k \sum_{i\in S_k} H(X_i \mid X_{A_{k-1}}) - H(X) \\ &= \sum_k \delta_k h(a_{k-1}) - H(X) = B_N. \end{aligned} \tag{19}$$

Similarly, (15) identifies C_N with the expected sum of per-token posterior KLs over revealed blocks. Both terms are nonnegative; only C_N depends on the denoiser or temperature.

Step 3: pass from paths to outputs. Apply the KL chain rule in the opposite order, first conditioning on the output X:

$$\mathrm{KL}(P \otimes \nu \,\|\, Q_\lambda) = \mathrm{KL}(P \,\|\, \pi_\lambda) + \mathbb{E}_{X\sim P}\,\mathrm{KL}(\nu \,\|\, Q_\lambda(S \mid X))\,. \tag{20}$$

Positive denoiser probabilities ensure $\pi_\lambda(x) > 0$ and $Q_\lambda(S \mid x) > 0$ for all relevant x, S. This proves (5). In particular,

$$\mathrm{KL}(P \,\|\, \pi_\lambda) \le B_N + C_N(\lambda), \qquad \mathrm{TV}(P, \pi_\lambda) \le \sqrt{\tfrac{1}{2}[B_N + C_N(\lambda)]}. \tag{21}$$

These are upper bounds. Neither the temperature-independence of B_N nor the convexity of C_N transfers automatically to output KL, because the subtracted conditional KL R_N depends on λ.

C.2.3 Consequences for parallel sampling

Dependence profile and step-size bounds. Applying (13) to each summand of h gives

$$\begin{aligned} D(a) := -h'(a) &= \sum_{i\neq j} \mathbb{E}_{A\subseteq[L]\setminus\{i,j\}}\left[H(X_i \mid X_A) - H(X_i \mid X_A, X_j)\right] \\ &= \sum_{i\neq j} \mathbb{E}_A I(X_i; X_j \mid X_A) \ge 0. \end{aligned} \tag{22}$$

The sum is over ordered pairs. Each other coordinate is independently included in A with probability a. This proves that h is nonincreasing, without claiming that its derivative has a fixed monotonicity or that h is convex.

On one grid interval $[b, d]$,

$$\begin{aligned} (d-b)h(b) - \int_b^d h(u)\,du = \int_b^d [h(b) - h(u)]\,du &= \int_b^d \int_b^u D(v)\,dv\,du \\ &= \int_b^d (d-v)D(v)\,dv. \end{aligned} \tag{23}$$

Summing gives the exact refinement of the parallelization term

$$B_N = \sum_{k=1}^{N} \int_{a_{k-1}}^{a_k} (a_k - u)D(u)\,du. \tag{24}$$

Writing $\Delta = \max_k \delta_k$ and using $D \ge 0$,

$$B_N \le \Delta \int_0^1 D(u)\,du = \Delta[h(0) - h(1)] = \Delta \sum_i I(X_i; X_{-i}). \tag{25}$$

The dependence sum equals total correlation plus dual total correlation:

$$\sum_i I(X_i; X_{-i}) = \underbrace{\sum_i H(X_i) - H(X)}_{\mathrm{TC}(X)} + \underbrace{H(X) - \sum_i H(X_i \mid X_{-i})}_{\mathrm{DTC}(X)}\,. \tag{26}$$

This connects the Bernoulli-mask profile to the information quantities in prior schedule analyses (Lavenant & Zanella, 2025; Chen et al., 2026).

For an equal grid $a_k = k/N$, let $T_N = \frac{1}{N}[\frac{1}{2}h(0) + \sum_{k=1}^{N-1} h(k/N) + \frac{1}{2}h(1)]$ be the composite trapezoidal rule. The left sum differs from T_N by $[h(0) - h(1)]/(2N)$. On any interval $[b, d]$, twice integrating by parts gives

$$\frac{d-b}{2}[h(b) + h(d)] - \int_b^d h(u)\,du = \frac{1}{2}\int_b^d (u-b)(d-u)h''(u)\,du. \tag{27}$$

The absolute value is bounded by $\|h''\|_\infty (d-b)^3/12$. Consequently,

$$\left| B_N - \frac{1}{2N}\sum_i I(X_i; X_{-i}) \right| \le \frac{\|h''\|_\infty}{12N^2}. \tag{28}$$

For fixed P and L, this supplies the asymptotic coefficient, not only an $O(N^{-1})$ rate. The constant depends on P, L; this is not a dimension-uniform expansion when L grows with N.

Inserting a grid point $b < c < d$ reduces its contribution to the left sum by $(d-c)[h(b) - h(c)] \geq 0$. Thus nested refinement never increases B_N. This statement concerns the parallelization term, not the learned-denoiser term C_N or the final output divergence. If e_λ is K-Lipschitz, its quadrature has the separate standard bound

$$\left| C_N(\lambda) - \int_0^1 e_\lambda(a)\, da \right| \leq \frac{K}{2} \sum_k \delta_k^2, \tag{29}$$

obtained by integrating $|e_\lambda(a_{k-1}) - e_\lambda(a)| \leq K(a - a_{k-1})$ on each interval.

Entropy inflation for two tokens. Let $L = 2$, let P be any law of (X_1, X_2) with marginals P_1, P_2, and let q be the Bayes denoiser at $\lambda = 1$. The two coordinates share a round with probability $c = \sum_k \delta_k^2$, in which case each is drawn from its marginal. Otherwise one is revealed first from its marginal and the other from its exact conditional, which reproduces P in either order. Hence $\pi_1 = c\, P_1 \otimes P_2 + (1-c)P$. By concavity of entropy,

$$H(\pi_1) \geq c\, H(P_1 \otimes P_2) + (1-c)H(P) = H(P) + c\, I(X_1; X_2). \tag{30}$$

Parallel sampling therefore adds at least $c\, I(X_1; X_2)$ nats of spurious randomness to a perfect model. Entropy inflation does not by itself fix the improving temperature direction: a few binary two-token targets prefer slight smoothing, which is why Theorem 2 fixes no direction. We do not know whether (30) extends to $L \geq 3$ under independent reveal rounds; $H(\pi_1) > H(P)$ held for all 280 random targets in Section C.3.

Reveal-round collisions. Under independent reveal rounds, the expected numbers of empty and singleton blocks, respectively, are

$$\sum_k (1-\delta_k)^L, \qquad \sum_k L\delta_k(1-\delta_k)^{L-1}. \tag{31}$$

A uniformly selected token shares its reveal round with another token with probability $1 - \sum_k \delta_k(1-\delta_k)^{L-1}$; the expected number of colliding unordered pairs is $\binom{L}{2} \sum_k \delta_k^2$. For $L = N = 1024$ and equal masses, the expected empty and singleton counts are 376.53 and 376.89, the sharing probability is $.63194$, and the expected colliding-pair count is 511.5. These describe sampling geometry, not a lower bound on output error or on actual model calls.

C.3 Proof of Theorem 2: the Bayes temperature is generically suboptimal

Part (a). If every round reveals at most one coordinate, a reveal path is an ordering σ of $[L]$, drawn independently of X. With Bayes conditionals, $Q_1^\sigma(x) = \prod_{j=1}^L P(x_{\sigma(j)} \mid x_{\sigma(1)}, \ldots, x_{\sigma(j-1)}) = P(x)$ by the chain rule, for every σ. Hence $\pi_1 = P$ and $\lambda = 1$ attains zero output KL. This includes left-to-right autoregressive decoding.

Part (b): one counterexample suffices. The derivative of output KL at $\lambda = 1$ is analytic in the target law. Showing it is nonzero for just one target therefore proves that it is nonzero almost everywhere. Parameterize full-support targets by the open simplex $\Delta^\circ \subset \mathbb{R}^{|\mathcal{V}|^L - 1}$, a connected open set. Each Bayes conditional $P(x_i \mid x_A) = P(x_i, x_A)/P(x_A)$ is a ratio of linear functions with a positive denominator, hence real-analytic and positive on Δ°. Tempering, $q_\lambda(v) = \exp(\lambda \log q(v)) / \sum_u \exp(\lambda \log q(u))$, is analytic in $(P, \lambda) \in \Delta^\circ \times (0, \infty)$. Each path law $Q_\lambda^S(x)$ is a finite product of such terms, $\pi_\lambda(x) = \sum_S \nu(S) Q_\lambda^S(x)$ is a positive finite sum, and $\mathrm{KL}(P \,\|\, \pi_\lambda) = \sum_x P(x)[\log P(x) - \log \pi_\lambda(x)]$ is therefore analytic. So is $g(P) = \partial_\lambda \mathrm{KL}(P \,\|\, \pi_\lambda)\,|_{\lambda=1}$. The zero set of a real-analytic function on a connected open set is either the whole set or Lebesgue-null (Mityagin, 2020). It is closed in Δ° because g is continuous. It remains to exhibit one target with $g \neq 0$ for every $L \geq 2$, $|\mathcal{V}| = V \geq 2$, and grid with $N \geq 2$.

A V-ary witness. Let X_1 be uniform on $\mathcal{V}$, let $X_2 = X_1$ with probability $s \in (1/V, 1)$ and otherwise be uniform on the other $V - 1$ symbols, and let $X_3, \ldots, X_L$ be i.i.d. uniform and

independent of (X_1, X_2). This target has full support, and X_2 is marginally uniform. Bayes conditionals of $X_3, \dots, X_L$ are uniform in every context and unchanged by tempering. The conditional of X_1 or X_2 is uniform when its partner is hidden, and otherwise the symmetric kernel $K(v \mid w) = s\,\mathbf{1}\{v = w\} + \frac{1-s}{V-1}\mathbf{1}\{v \neq w\}$, whose tempered version predicts agreement with probability

$$u_\lambda = \frac{s^\lambda}{s^\lambda + (V-1)^{1-\lambda}(1-s)^\lambda}. \tag{32}$$

The pair shares a round with probability $c = \sum_k \delta_k^2 \in (0, 1)$, since $N \geq 2$ and every $\delta_k > 0$. In that case both coordinates are uniform and independent; otherwise one is uniform and the other follows the tempered kernel, which is symmetric, so the order does not matter. Consequently π_λ keeps $X_3, \dots, X_L$ uniform and independent, keeps X_1 uniform, and makes the pair agree with probability $t_\lambda = c/V + (1-c)u_\lambda$, uniformly over agreeing and over disagreeing pairs. P has the same structure with agreement probability s, so $\mathrm{KL}(P \,\|\, \pi_\lambda) = \mathrm{KL}(\mathrm{Bern}(s) \,\|\, \mathrm{Bern}(t_\lambda))$ and

$$g(P) = \frac{t_1 - s}{t_1(1-t_1)}\,(1-c)\,u_1(1-u_1)\log\frac{s(V-1)}{1-s}, \qquad t_1 - s = c\Big(\frac{1}{V} - s\Big). \tag{33}$$

For $s \in (1/V, 1)$ the first factor is negative and the logarithm positive, so $g(P) < 0$: sharpening strictly improves the Bayes sampler. For $V = 2$ this is the family of Proposition 2. This completes the proof.

An explicit form of the derivative. For a path S and output x, let $A_i(S)$ be the coordinates revealed before the round of i, and define the path surprisal $\sigma(x, S) = -\log Q_1^S(x) = \sum_i -\log P(x_i \mid x_{A_i(S)})$, the path entropy $\eta(x, S) = \sum_i H(X_i \mid X_{A_i(S)} = x_{A_i(S)})$, and the path weight $w(x, S) = Q_1^S(x)/\pi_1(x) = Q_1(S \mid x)/\nu(S)$. Since $\partial_\lambda \log q_\lambda(v)|_{\lambda=1} = \log q(v) + H(q)$, we have $\partial_\lambda \log Q_\lambda^S(x)|_{\lambda=1} = \eta(x, S) - \sigma(x, S)$, and differentiating $\mathrm{KL}(P \,\|\, \pi_\lambda) = \sum_x P(x)\log P(x) - \sum_x P(x)\log\sum_S \nu(S)Q_\lambda^S(x)$ gives

$$g(P) = \mathbb{E}_{P\otimes\nu}\big[w\,(\sigma - \eta)\big] = \mathrm{Cov}_{P\otimes\nu}(w, \sigma - \eta). \tag{34}$$

The last equality uses $\mathbb{E}_\nu w(x, \cdot) = 1$ for every x and $\mathbb{E}_{P\otimes\nu}[\sigma - \eta] = 0$, which holds because, under $P \otimes \nu$, each X_i given $X_{A_i(S)}$ follows its Bayes conditional. For fixed x, $w = e^{-\sigma}/\pi_1(x)$ is decreasing in σ, so $\mathbb{E}_P \mathrm{Cov}_\nu(w, \sigma \mid X) \leq 0$ and

$$g(P) = \mathbb{E}_P \mathrm{Cov}_\nu(w, \sigma \mid X) - \mathrm{Cov}_{P\otimes\nu}(w, \eta). \tag{35}$$

The first term always pushes toward sharpening: the sampler over-weights the paths along which the data look least surprising. Reversing the direction requires a sufficiently negative covariance between path weight and conditional entropy: the favored paths must have lower, rather than higher, conditional entropy. $\lambda = 1$ is stationary exactly when the two terms cancel.

Random-target enumeration. The supplementary verifier enumerates every output and reveal assignment for $(L, |\mathcal{V}|, N) \in \{(2,2,2), (2,3,2), (2,2,3), (3,2,2), (3,2,3), (2,4,2), (3,3,2)\}$, 40 targets each, drawn from symmetric Dirichlet laws with concentration in $\{.3, 1, 3\}$, with random grids. For each target it checks the accounting identity of Theorem 1, compares (34) with finite differences, confirms that $\lambda = 1$ is not stationary, and locates the output-optimal $\lambda_\star$ on a logarithmic grid refined by bounded search. It also checks (30), the V-ary witness in closed form, part (ii) of Proposition 2, part (a) of Theorem 2, and Proposition 1 on random linear programs: 2,937 assertions in all, with maximum residual 4×10^{-11}. Sharpening is the improving direction for 279 of the 280 targets. The median $\lambda_\star$ is 1.16 (5th–95th percentiles 1.003–1.69), and the median relative reduction in output KL at $\lambda_\star$ is 5.6% (95th percentile 39%). These are properties of small random targets, not estimates for language.

C.4 Proof of Proposition 2: exact repair and inverted rankings

The two branches and their losses. For P_s in (6), both one-coordinate marginals are uniform. The Bayes probability that the second token matches the first, conditional on the first being visible, is s. Raising the two conditional probabilities to power λ and normalizing yields

$$u_\lambda = \frac{s^\lambda}{s^\lambda + (1-s)^\lambda} = \mathrm{sigmoid}(\lambda\,\mathrm{logit}\,s). \tag{36}$$

With probability $c = \sum_k \delta_k^2$, both coordinates reveal in the same round and are drawn independently from uniform marginals. They then match with probability $1/2$. Otherwise, the first token is uniform and the second matches it with probability u_λ. Both branches are invariant under swapping coordinates and flipping both bits. Their mixture is therefore exactly P_{t_λ} with $t_\lambda = c/2 + (1-c)u_\lambda$. Summing the two diagonal and two off-diagonal KL contributions proves the first identity in (7).

The unconditional marginals contribute no posterior error at any temperature. On a separate-round path, only the second token contributes error, namely $\mathrm{KL}(\mathrm{Bern}(s) \,\|\, \mathrm{Bern}(u_\lambda))$; this path has probability $1-c$. This proves the second identity. Since $s > 1/2$, u_λ is strictly increasing in λ and equals s exactly at $\lambda = 1$. Hence C_N is uniquely minimized at 1 for $c < 1$. The parallelization term is also explicit:

$$B_N = c\,[\log 2 - H_{\mathrm{b}}(s)], \qquad H_{\mathrm{b}}(s) = -s \log s - (1-s)\log(1-s). \tag{37}$$

(i) Exact repair. For the output objective, differentiation gives

$$\frac{d}{d\lambda}\,\mathrm{KL}(\mathrm{Bern}(s) \,\|\, \mathrm{Bern}(t_\lambda)) = \frac{t_\lambda - s}{t_\lambda(1-t_\lambda)}(1-c)u_\lambda(1-u_\lambda)\,\mathrm{logit}\, s. \tag{38}$$

All factors other than $t_\lambda - s$ are positive. As λ increases from 0 to infinity, t_λ strictly increases from $1/2$ to $1 - c/2$ (the latter not attained at finite λ).

If $c < 2(1-s)$, the target s lies strictly between these endpoints. There is a unique zero of the derivative, at $u_* = (s - c/2)/(1-c)$. It satisfies $u_* - s = c(s-1/2)/(1-c) > 0$, hence $\lambda_* > 1$, and inverting the logistic map gives the expression in Proposition 2(i). At that value $\pi_{\lambda_*} = P_s$, giving zero output KL and strictly positive $C_N(\lambda_*)$.

If $c \geq 2(1-s)$, then $t_\lambda < s$ for every finite λ. The derivative is strictly negative, and the infimum occurs only at $\lambda \to \infty$. The limiting error is zero when $c = 2(1-s)$ and strictly positive when $c > 2(1-s)$. This proves both regimes. The excluded case $c = 1$ reveals both coordinates together with probability one; temperature leaves the uniform product output unchanged and there is no strict posterior optimum.

(ii) Inverted ranking. In the finite-repair regime, consider any denoiser that, like the Bayes denoiser, predicts a uniform bit when the partner is hidden and predicts agreement with probability u when the partner is visible. The argument above, with u_λ replaced by u, gives output law $P_{t(u)}$ with $t(u) = c/2 + (1-c)u$ and posterior loss $(1-c)\,\mathrm{KL}(\mathrm{Bern}(s) \,\|\, \mathrm{Bern}(u))$, which is strictly positive for $u \neq s$. Because $t \mapsto \mathrm{KL}(\mathrm{Bern}(s) \,\|\, \mathrm{Bern}(t))$ strictly decreases on $(0, s]$, and $t(s) < t(u) \leq t(u_\star) = s$ for $u \in (s, u_\star]$, every such denoiser has strictly smaller output error than the Bayes denoiser at $\lambda = 1$ and strictly larger posterior loss. Tempering either denoiser sweeps its agreement probability continuously over $(1/2, 1)$, so both reach $u_\star$ and exactness at their own best temperatures; the inversion is a property of the shared fixed temperature.

(iii) Vanishing with steps. For fixed s, the finite-repair regime holds for all sufficiently small c. Taylor expansion of the explicit expression yields

$$\lambda_* = 1 + \frac{s - 1/2}{s(1-s)\,\mathrm{logit}\, s}\, c + O(c^2), \qquad c \to 0. \tag{39}$$

For an equal N-round grid, $c = 1/N$, so the output-optimal temperature tends to the Bayes value as collisions vanish. This is a property of the stated family, not a temperature schedule prescribed for language models.

Stability under perturbations. At any point in the finite-repair regime, fix the value $\lambda_* > 1$. Its output KL is strictly smaller than at $\lambda = 1$. On the interior of the finite probability simplex, Bayes conditionals, their tempered probabilities, the finite mixture output, and its KL are continuous functions of the target probabilities and grid masses. The strict improvement therefore persists in a neighborhood of that target and schedule. Exact zero-error repair relies on the symmetric family, but the existence of a sharpening benefit for a Bayes denoiser is stable to sufficiently small perturbations.

C.5 Posterior-loss measurement and implementation

We derive the estimator used in Section 4.4, then relate the ideal reveal grid to the implemented sampler.

C.5.1 THE FORCED-MASK ESTIMATOR AND TEMPERATURE DERIVATIVES

Draw $X \sim P$, $K \sim \mathrm{Cat}(\delta)$, and $I \sim \mathrm{Unif}([L])$, independently. Draw $A \subseteq [L] \setminus \{I\}$ by independent inclusion at probability a_{K-1}. The unbiased estimator is

$$\widehat{\mathcal{J}}_N = L[-\log q^I_\lambda(X_I \mid X_A, a_{K-1})], \qquad \mathbb{E}\widehat{\mathcal{J}}_N = \mathcal{J}_N(\lambda). \tag{40}$$

The factor L cancels the uniform coordinate choice:

$$\mathbb{E}\{L[-\log q^I_\lambda(X_I \mid X_A, a_{K-1})]\} = \sum_k \delta_k \sum_i \mathbb{E}_{A,X}[-\log q^i_\lambda(X_i \mid X_A, a_{k-1})] = \mathcal{J}_N(\lambda). \tag{41}$$

The target coordinate is always masked by construction. No inverse factor $1/(1-a)$ is needed. This establishes unbiasedness, not a universal variance comparison with every alternative estimator.

Temperature scaling is convex in inverse temperature for fixed logits (Guo et al., 2017); here those contexts are weighted by the reveal grid. For a fixed context and target x, let $r(v)$ be finite logits and $q_\lambda(v) = \exp(\lambda r(v))/\sum_u \exp(\lambda r(u))$. Then

$$\frac{d}{d\lambda}[-\log q_\lambda(x)] = \mathbb{E}_{Y\sim q_\lambda} r(Y) - r(x), \tag{42}$$

$$\frac{d^2}{d\lambda^2}[-\log q_\lambda(x)] = \mathrm{Var}_{Y\sim q_\lambda}\, r(Y) \geq 0. \tag{43}$$

At a finite grid there are finitely many contexts, so differentiation commutes with their weighted sum. Therefore

$$\mathcal{J}'_N(\lambda) = L\mathbb{E}[\mathbb{E}_{Y\sim q^I_\lambda} r^I(Y) - r^I(X_I)], \qquad \mathcal{J}''_N(\lambda) = L\mathbb{E}\,\mathrm{Var}_{Y\sim q^I_\lambda}\, r^I(Y). \tag{44}$$

The first expression is also an unbiased forced-mask gradient estimator. If at least one positively weighted context has nonconstant logits, its variance is strictly positive at every finite λ, because all token probabilities are positive. Then $\mathcal{J}_N$ is strictly convex. A minimizer exists on every nonempty compact inverse-temperature interval and is unique under this strictness condition. An unconstrained finite minimizer on $(0, \infty)$ is not guaranteed.

For an exact Bayes denoiser, $C_N(1) = 0$, so $\lambda = 1$ minimizes $\mathcal{J}_N$. It is unique when a positively weighted Bayes conditional is nonuniform (and positive). Conversely, a negative derivative $\mathcal{J}'_N(1)$ supports sufficiently small increases in inverse temperature, not an arbitrary finite sharpening. A positive measured finite difference $\mathcal{J}_N(1/.9) - \mathcal{J}_N(1)$ does not, by itself, determine the derivative at 1. Our empirical statement is the finite contrast, not a claim to exclude every possible infinitesimal calibration adjustment.

C.5.2 SCHEDULES, PRECISION, AND VERIFICATION

Configured updates and endpoint completion. The empirical MDLM code uses a clipped linear schedule, starts from all masks, and applies M updates followed by terminal cleanup at $\epsilon = 10^{-5}$. The code uses $\alpha(t) = 1 - (1-\eta)t$, where $\eta = 10^{-3}$ is the schedule clipping constant, distinct from the terminal time ϵ. For its idealized categorical counterpart, a masked token survives an update $t \to s$ with probability $(1-\alpha(s))/(1-\alpha(t)) = s/t$. Starting from all masks at $t = 1$, its survival probability at time t is therefore t. The corresponding visible probability is $a = 1-t$, rather than $\alpha(t)$. A uniform time grid from 1 to ϵ consequently gives M ordinary reveal masses $(1-\epsilon)/M$ and terminal mass ϵ, so $N = M + 1$. The time-independent MDLM predictor allows this reparameterization without changing its logits. The proper-loss measurement uses these context weights; it does not measure the fp32 output kernel's KL. Probability floors, roundoff, and categorical implementation effects are not included in the exact law above.

If two procedures have *identical* laws before terminal cleanup and differ only in how they fill surviving masks, coupling their common prefix gives output-TV difference at most $\Pr(\text{some mask survives}) = 1 - (1-\epsilon)^L \leq L\epsilon$. For $L = 1024$ this is approximately .01019. This comparison does not justify replacing a whole earlier grid by a different grid: their prefix laws need not agree.

Categorical precision. The implementation uses one categorical draw over clean tokens and the mask outcome. Finite precision can therefore change both token probabilities and reveal timing; the

independent reveal law ν above describes the exact-arithmetic counterpart. The following comparison treats these changes jointly as perturbations of the complete transition kernel. Suppose exact and implemented transition kernels P_k and $\widetilde{P}_k$ obey $\sup_z \mathrm{TV}(P_k(z,\cdot), \widetilde{P}_k(z,\cdot)) \le \xi_k$ with $\xi_k \in [0,1]$ at every step, including cleanup. A stepwise maximal coupling, conditional on no previous disagreement, succeeds at step k with probability at least $1-\xi_k$. Thus

$$\mathrm{TV}(\pi, \widetilde{\pi}) \le 1 - \prod_k (1-\xi_k) \le \sum_k \xi_k. \tag{45}$$

No numerical value of ξ_k is assumed or estimated here. Finite arithmetic may alter support, so a TV perturbation statement cannot be substituted for a finite forward-KL guarantee. In particular, fp32 is an experimental operating point, not a mathematically exact implementation of a universal lower temperature.

Verification artifacts. The supplementary verifiers enumerate tiny binary and ternary distributions, including distributions with zero-probability sequences, and nonuniform reveal grids. They check augmented and marginal KL identities, conditional total correlation, convex-temperature derivatives, finite-grid bounds, both regimes and the equality boundary of the collision example, stability under asymmetric target perturbations, the clipped-schedule reveal masses, and the random-target results of Section C.3. Separate integer/Fraction coefficient checks verify the entropy-area and conditional-mutual-information identities for formal subset entropies through $L = 8$. These checks complement the proofs above; they are not proof-assistant certification.

D Extended Related Work and Implications for Sampler Design

Our findings connect three research questions. How much of the few-step gap requires a new model? Which sampling changes can close it? And which measurements can tell whether it has closed? We develop these connections below, extending the recommendations in Section 5.

D.1 Stronger baselines and the few-step gap

Few-step quality of masked diffusion (Austin et al., 2021; Campbell et al., 2022; Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2025) has been pursued through distillation (Deschenaux & Gulcehre, 2025; Hayakawa et al., 2025; Sahoo et al., 2025; Zheng et al., 2026), consistency training (Amin et al., 2026), and alternative processes (Lou et al., 2024; von Rütte et al., 2025; Pynadath et al., 2026). These methods change the trained generator. Our audit asks how much the original checkpoint can already do when its sampler is tuned.

For language, Zheng et al. (2025) identified fp32 truncation and advocated fp64. CANDI showed that temperature can trade for steps along perplexity–entropy frontiers (Pynadath et al., 2026), and CaRE found that temperature explains most MAUVE variance among remasking strategies (Shah et al., 2026). Our results establish that the tuned baseline rivals its distilled successors, that the gain survives semantic evaluation, and that a departure from temperature one helps even exact posteriors under parallel sampling. The public MDLM moves from gPPL 107.1 at 1,024 standard steps to 41.3 at 64 sharpened steps (Table 1). At 256 steps it also exceeds the standard sampler at 1,024 on GEQ under every judge (Section 3.3). The baseline correction thus survives a change of measurement.

Related evidence comes from exact test-time likelihoods, which find MDMs better than previously thought (Turok et al., 2026), and from analyses tracing much of the masked–uniform gap to parameterization and sampling design (Gourevitch et al., 2026). Tuning baselines has closed apparent gaps before, for GANs (Lucic et al., 2018) and neural language models (Melis et al., 2018). Here the correction also exposes an inversion between predictor accuracy and generation quality.

D.2 Sampler design and the price of parallelism

A growing literature improves pretrained diffusion LMs at inference alone. Theory bounds factorization error through information profiles and total correlation (Lavenant & Zanella, 2025; Wainwright, 2026), characterizes it exactly for any schedule (Chen et al., 2026), relates it to generation order (Zhang et al., 2026), and shows that it can escape likelihood-based metrics (Tang et al., 2026).

Theorem 1 extends the accounting to learned, tempered denoisers and isolates the temperature-dependent credit for mixing paths. This separates improving posterior calibration (Platt, 2000; Guo et al., 2017) from improving the generated distribution.

One accounting, several ways to improve the sampler. For non-adaptive reveal rounds, output error is $B_N + C_N(\lambda) - R_N(\lambda)$. This decomposition distinguishes the mechanisms that sampler design can exploit.

- **Schedule.** B_N depends only on the data and grid. Its mesh bound (25) and the fact that nested refinement never increases it (Section C.2.3) explain why schedule design (Lavenant & Zanella, 2025; Chen et al., 2026; Wainwright, 2026) helps.
- **Temperature and truncation.** Exact temperature scaling leaves the reveal law and B_N unchanged and trades posterior error C_N against path credit R_N (Proposition 2). Truncation can also sharpen predictions. Finite-precision categorical draws can additionally change reveal timing, so fp32 is not equivalent to a scalar temperature adjustment (Section C.5.2).
- **Order and dependence-aware selection.** Samplers plan or adapt the unmasking order (Liu et al., 2025; Kim et al., 2025a; Ben-Hamu et al., 2025; Hayakawa et al., 2026), or space and screen reveal sets for dependence (Luxembourg et al., 2026; Azangulov et al., 2026; Sahin et al., 2026; Kim et al., 2025b). They target the same-round dependence that B_N prices. State-dependent reveal laws fall outside Theorem 1 as stated.
- **Joint draws.** Learned joint samplers (Bansal & Sanghavi, 2025) and mixture kernels (Ahmed et al., 2026) model dependence among same-round tokens, addressing the factorization that gives rise to B_N.
- **Revision.** Remasking and predictor–corrector samplers (Wang et al., 2025; Deschenaux et al., 2026) revisit committed tokens, which also lies outside our accounting.

These approaches can complement sharpening. Their gains should be measured after controlling the baseline's sharpness, so that a new mechanism receives credit for what it adds.

Temperature confounds sampler comparisons. Sharpening alone substantially improves the baseline, and temperature one is generically suboptimal even for an exact denoiser under parallel sampling (Theorem 2). Some samplers also change sharpness implicitly. Confidence-based selection carries an implicit temperature (Hayakawa et al., 2026), and fp32 draws truncate categorical noise (Zheng et al., 2025). A comparison against a baseline at $\tau = 1$ can therefore credit a new sampler with a temperature gain. Four practices help separate the effects.

1. Tune temperature and precision for both the baseline and proposed sampler under the same screen, disclose the search budget, and report the gain over the baseline's best setting.
2. Report configured steps and network calls separately, together with measured latency and memory. Caching and adaptive reveal can change their relationship (Section A.1).
3. Supplement gPPL and unigram entropy with group-level evaluation. Proposition 1 shows why matching token entropy does not rule out collapse.
4. Evaluate checkpoints together with their samplers. Held-out posterior loss sees $B_N + C_N$ but not R_N (Section 4.4). Exact test-time likelihood, where available (Turok et al., 2026), instead scores the output law itself.

For the tested 8B diffusion LMs decoded greedily, positive temperatures do not help (Section A.8). The temperature comparison is most relevant to stochastic sampling. Under greedy decoding, order, selection, and revision remain available, and our analysis does not attribute a particular adaptive sampler's gain to temperature.

D.3 Evaluating quality and diversity across outputs

Quality–diversity trade-offs and the limits of surface metrics are well documented (Hashimoto et al., 2019; Holtzman et al., 2020; Tevet & Berant, 2021). Reference-based (Pillutla et al., 2021), kernel (Friedman & Dieng, 2023), meaning-cluster (Kuhn et al., 2023), and distinct-response (Zhang et al., 2025) measures capture related notions. LLM judges provide scalable but biased ratings (Liu

et al., 2023; Zheng et al., 2023; Wang et al., 2024). GroupEval combines blinded group-level mode partitioning, validity-weighted entropy, and a separate quality score for unconditional generation, where no prompt defines equivalence. Its contribution is a fixed, reusable protocol and the research conclusions it changes.

Likelihood is known to favor repetitive text (Holtzman et al., 2020). The DiDi comparison shows that token entropy need not detect this failure, even when two likelihood evaluators agree. GroupEval also distinguishes between sharpening gains that survive semantic judgment and those that do not. Its quality and diversity scores remain separate because good individual outputs and a varied output distribution are distinct requirements. Human calibration and validation of the mode assignments remain open, while the released rubric, grouping rules, and scorer make this distinction directly testable under a common protocol.