Gentoo: Draft Continuously, Decode Discretely

Anonymous authors
Paper under double-blind review

ABSTRACT

Masked diffusion models speed up generation by committing multiple tokens per round, each sampled independently from its predicted distribution. Tokens committed together can therefore conflict, especially with fewer rounds. To address these conflicts, some methods select which tokens to commit together or jointly denoise auxiliary continuous variables with the tokens, but they still settle instance content during token commitment. Our key insight is therefore to draft continuously and decode discretely: a fine-grained, well-structured draft gives parallel token predictions a common reference before the first commitment. Gentoo first uses a pretrained latent generator to draft an instance in representation space. A learned adapter then provides this draft as shared context to a pretrained masked diffusion decoder, which translates it into tokens and progressively completes the sequence. To align training encodings of real samples with drafts generated at inference, we partially noise each encoding and let the latent generator denoise it, while keeping the sample's own tokens as targets. With the shared draft, tokens committed in the same round become more consistent, and quality depends far less on the number of rounds. Gentoo improves generation quality over MaskBit and Meissonic while using only 62.5% and 64% of their respective original NFE budgets, counting both continuous and discrete stages. These configurations lower inference time by 16% on ImageNet and 53% in text-to-image generation.

1 INTRODUCTION

Masked discrete diffusion has become a standard route to parallel generation of discrete data: a bidirectional decoder predicts every masked position from the visible ones and commits several positions per round, for images represented as codebook tokens (Chang et al., 2022; Weber et al., 2024; Chang et al., 2023; Bai et al., 2024) and for text (Lou et al., 2023; Sahoo et al., 2024; Shi et al., 2024). The decoder reads the whole sequence, so its hidden states can model dependencies among positions, but its output factorizes into per-token distributions, and positions committed in the same round are sampled independently. If two masked positions jointly form "New York" or "San Diego" with equal probability, independent sampling yields "New Diego" or "San York" half the time. Prior analyses characterize such inconsistency as the factorization error of parallel decoding (Liu et al., 2025; Shariatian et al., 2025; Zhang et al., 2026b): it grows with the number of tokens committed per round, and because standard samplers do not revise committed tokens, the errors propagate to all later predictions, so generation quality drops sharply as the number of rounds is reduced.

Existing approaches address this difficulty through token sampling or auxiliary continuous context. Sampler-side methods select compatible tokens to commit together (Azangulov et al., 2026; Ringel et al., 2026) or correct factorized predictions with a joint model (Xu et al., 2025; Liu et al., 2025), trading parallelism or additional computation for more consistent commits. Continuous-context methods follow two designs. One jointly denoises the continuous variables and discrete tokens, allowing them to interact throughout generation, but leaving the continuous context unsettled when early tokens are committed (Zheng et al., 2026c; Pynadath et al., 2025; Zhou et al., 2025; Shariatian et al., 2025). The other first samples a representation from a prior and then generates tokens conditioned on it, as in RCG and sequential LADD (Li et al., 2024a; Shariatian et al., 2025). This provides a common reference before the first commitment, but requires balancing the information it carries against the difficulty of generating it. RCG uses a lightweight prior over a single 256-dimensional global vector, supplying semantic context without an explicit patchwise draft (Li et al., 2024a). Sequential LADD encounters the other side of this tradeoff: its informative continuous latent becomes a generative bottleneck, leading the authors to omit the continuous sequential variant from their text experiments (Shariatian et al., 2025, Remark 5.1). This motivates our question: can

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

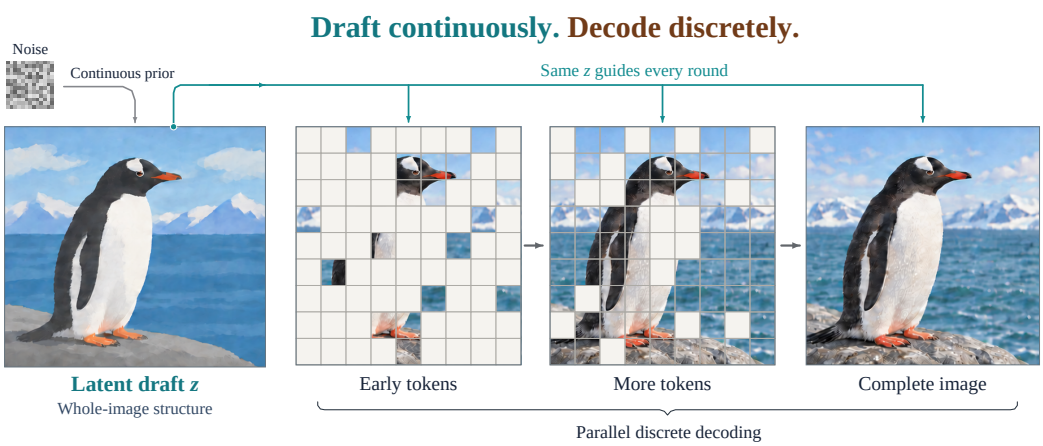

Figure 1: Draft continuously, decode discretely. In Gentoo, a continuous prior samples an encoder-space representation z that conditions every discrete decoding round. The whole-image draft and progressively filled token panels visualize this process schematically.

we first generate a fine-grained, well-structured instance draft in continuous space, giving parallel predictions a sufficiently detailed common reference before any token is committed?

Gentoo answers this question by drafting continuously and decoding discretely. A pretrained continuous drafter in representation space first forms a fine-grained, well-structured instance draft, using continuous denoising to jointly refine its content before any discrete commitment (Zheng et al., 2026a; Tong et al., 2026; Shariatian et al., 2025). A learned adapter maps the draft into context tokens that a pretrained masked decoder attends to throughout generation. Under this shared context, the decoder translates the draft into an initially incomplete sequence of discrete tokens, then progressively fills the remaining positions using both the draft and the tokens already committed. Translation and completion occur within the same masked decoding process, with the continuous draft held fixed across rounds. This division of work gives parallel token predictions a common instance reference, allowing the discrete stage to concentrate on realizing the draft with fewer rounds.

To improve how the decoder translates generated drafts, Gentoo uses *drafter-refined conditioning*. Each training image provides a continuous encoding and its corresponding discrete tokens. We partially noise the encoding and let the drafter denoise it, then train the masked decoder to predict the image's original tokens from the refined representation. Varying the noise level controls how much instance information is retained in the starting state and how much the drafter can reshape it. The decoder thus learns from conditions transformed by the same drafter that supplies drafts at inference, while retaining the original paired token supervision.

We evaluate Gentoo on class-conditional ImageNet generation and text-to-image generation. Counting both drafting and decoding, Gentoo improves generation quality over MaskBit and Meissonic using only 62.5% and 64% of their respective original NFE budgets. These configurations reduce ImageNet generation time by 16% and the sum of measured text-to-image component runtimes by 53%. Our contributions are:

- **Characterizing parallel coordination.** We apply a total-correlation identity to characterize how shared context changes the dependence cost of parallel decoding. Complementary diagnostics on held-out tokens and actual generation trajectories find more compatible predictions when instance context is supplied.
- **Drafter-refined paired conditioning.** We construct training conditions by partially noising encoded representations and denoising them with the continuous drafter. Their original tokens remain the supervision targets, teaching the decoder to interpret drafter-refined representations and improving generation over training on clean encodings.
- **Draft continuously, decode discretely.** Gentoo connects an independently pretrained continuous drafter and masked decoder through shared instance context. The decoder translates the draft into tokens and progressively completes the sequence. This improves quality with fewer total sampling updates and lower inference cost.

2 Related Work

Masked and discrete generation. Masked diffusion (Austin et al., 2021; Lou et al., 2023; Sahoo et al., 2024; Shi et al., 2024) supports parallel generation, with advances in image-token decoding (Chang et al., 2022), representation learning and tokenization (Li et al., 2023; Weber et al., 2024; Yu et al., 2024a;b), and text-to-image synthesis (Chang et al., 2023; Xie et al., 2025; Kim et al., 2025; Bai et al., 2024). Parallel factorization incurs dependence costs (Liu et al., 2025; Zhang et al., 2026b). PUNT and DEMASK select parallel updates by testing or predicting the effects of revealing tokens (Azangulov et al., 2026; Ringel et al., 2026), while energy and copula models correct joint proposals (Xu et al., 2025; Liu et al., 2025). Other approaches improve generation through reveal schedules and token critics (Besnier et al., 2025; Lezama et al., 2022), remasking (Wang et al., 2026a), distillation (Deschenaux & Gulcehre, 2025; Hayakawa et al., 2024; Sahoo et al., 2025; Zheng et al., 2026b; Zhu et al., 2025; 2026), and reward post-training (Zhang et al., 2026a). Gentoo supplies shared instance context while retaining the decoder's vocabulary, objective, and sampler.

Continuous variables and representation priors. Continuous augmentation uses noisy channels (Zheng et al., 2026c; Pynadath et al., 2025), variational latents (Xie et al., 2026), or stochastic embeddings and self-conditioning (Yoo et al., 2026; Cardei et al., 2026). CCDD jointly evolves tokens and encoder latents (Zhou et al., 2025), while LADD studies joint and sequential generation (Shariatian et al., 2025). Other work combines continuous generation with conditional text decoding (Guo et al., 2026; Shen et al., 2026), distills latent priors (Lemercier et al., 2026), or uses dependency graphs and quantized representations (Liang et al., 2026; Wang et al., 2026b). Representation-conditioned generation (Preechakul et al., 2022; Ramesh et al., 2022) includes RCG, which generates self-supervised representations to condition generators including MAGE's masked decoder (Li et al., 2024a). Gentoo pairs independently pretrained representation priors (Zheng et al., 2026a; Tong et al., 2026) with adapted masked decoders, sampling one fine-grained, well-structured draft before decoding and sharing it across rounds. Drawing on partial noising and denoising (Meng et al., 2021) and conditioning augmentation (Ho et al., 2022; Shabalin et al., 2025; Shen et al., 2026), Gentoo refines training encodings through the prior while retaining paired token targets.

3 Method

Gentoo drafts continuously and decodes discretely (Figure 2). A continuous prior first generates an instance draft in representation space; a learned adapter exposes it as shared context, through which a masked decoder translates the draft into tokens and progressively completes the sequence. The same draft conditions every decoding round, giving parallel predictions a common reference from the first commitment. We first analyze the role of shared context in parallel commits, then specify the continuous-to-discrete handoff and its sampling budgets, and finally describe how we train a pretrained decoder to read generated drafts.

3.1 What a shared instance context buys for parallel commits

The cost of a parallel commit. Let $X = T(Y) \in \mathcal{V}^N$ denote an instance's tokens under a fixed tokenizer, and C an optional class or text condition. In one decoding round, V contains the visible positions and $B \subseteq [N] \setminus V$ is the block committed together. Tokens in B are sampled independently,

$$q(x_B \mid x_V, c) = \prod_{i \in B} q_i(x_i \mid x_V, c). \tag{1}$$

We analyze fixed blocks under the data distribution with exact conditionals, isolating the cost of this factorization from prediction error. Even when each marginal is correct, their product need not recover the joint conditional. The resulting divergence is the conditional total correlation (Liu et al., 2025; Shariatian et al., 2025),

$$\begin{aligned} \mathrm{TC}(X_B \mid X_V, C) &:= \mathbb{E}_{X_V, C}\left[\mathrm{KL}\left(p(X_B \mid X_V, C) \,\middle\|\, \prod_{i \in B} p(X_i \mid X_V, C)\right)\right] \\ &= \sum_{i \in B} \mathrm{H}(X_i \mid X_V, C) - \mathrm{H}(X_B \mid X_V, C). \end{aligned} \tag{2}$$

Here H denotes conditional entropy. This cost vanishes exactly when the positions in B are conditionally independent given (X_V, C). Otherwise, parallel sampling discards dependence that sequential sampling can recover by conditioning on earlier commitments. This is a structural cost of the factorized output distribution: hidden states can model dependencies, but independently sampling even exact marginals does not recover the joint. Adding a position $j \notin B \cup V$ increases the cost by $\mathrm{I}(X_j; X_B \mid X_V, C) \geq 0$, explaining why larger parallel commits expose more dependence.

A shared instance context. An auxiliary context Z changes this cost to $\mathrm{TC}(X_B \mid X_V, Z, C)$, connecting context conditioning to the dependence cost of parallel generation (Shariatian et al., 2025). Applying the total-correlation explanation identity of Ver Steeg & Galstyan (2014) gives:

Proposition 1 *For any Z jointly distributed with (X, C) and fixed sets V and $B \subseteq [N] \setminus V$,*

$$\begin{aligned} \mathcal{S}_Z(B \mid V) &:= \mathrm{TC}(X_B \mid X_V, C) - \mathrm{TC}(X_B \mid X_V, Z, C) \\ &= \sum_{i \in B} \mathrm{I}(X_i; Z \mid X_V, C) - \mathrm{I}(X_B; Z \mid X_V, C). \end{aligned} \tag{3}$$

Subtracting the two entropy expansions in (2) proves the identity: each difference $\mathrm{H}(X_A \mid X_V, C) - \mathrm{H}(X_A \mid X_V, Z, C)$ equals $\mathrm{I}(X_A; Z \mid X_V, C)$, for $A = B$ or $A = \{i\}$. Here I denotes conditional mutual information. The quantity $\mathcal{S}_Z(B \mid V)$ measures the dependence explained by the shared context: positive values indicate lower factorization cost, while negative values indicate increased residual dependence. Thus, the benefit depends on what information the context supplies across the committed positions. Applying the mutual-information chain rule across a fixed decoding schedule gives the following corollary.

Corollary 1 *For disjoint blocks $B_1, \ldots, B_R$ partitioning $[N]$, with $V_r = \bigcup_{\ell < r} B_\ell$,*

$$\sum_{r=1}^{R} \mathcal{S}_Z(B_r \mid V_r) = \sum_{r=1}^{R} \sum_{i \in B_r} \mathrm{I}(X_i; Z \mid X_{V_r}, C) - \mathrm{I}(X; Z \mid C). \tag{4}$$

Summing (3) over rounds yields this result because the chain rule gives $\sum_r \mathrm{I}(X_{B_r}; Z \mid X_{V_r}, C) = \mathrm{I}(X; Z \mid C)$. The cumulative reduction is zero for single-token commits. For one-round decoding, it becomes $\sum_i \mathrm{I}(X_i; Z \mid C) - \mathrm{I}(X; Z \mid C)$. At intermediate budgets, it depends on the committed blocks and their visible contexts. Proofs are given in Appendix D.

From shared context to continuous drafts. This analysis motivates establishing instance information shared across positions before the first commitment. To draft continuously and decode discretely, Gentoo uses $Z = E(Y)$, the representation of a pretrained encoder, as the interface between the two stages. Each training instance supplies both a representation and its target tokens, while a representation prior supplies drafts at inference. The draft remains available as visible tokens accumulate, giving every round access to the same instance reference. We assess its effect on learned predictions with complementary diagnostics on generated trajectories and held-out images.

Measurement. On generated trajectories, *commit conflict* randomly splits each committed block, reveals one half, and measures the drop in log probability of sampled tokens in the other half, then reverses the roles. Both directions start from the same pre-commit canvas and leave the trajectory unchanged. With exact conditionals and a fixed block, its expectation is the mean KL divergence from each token's pre-reveal distribution to its distribution after observing the other half; lower conflict indicates more compatible commitments. On held-out images, the *CTC diagnostic* compares parallel NLL with a teacher-forced chain that reveals preceding target tokens. Its expected gap equals conditional total correlation per component with exact conditionals, but learned probabilities also contribute approximation error. Disabling context or shuffling it within a class tests whether the reduction depends on instance-specific information rather than class identity alone.

3.2 Draft Continuously, Decode Discretely

Gentoo drafts continuously, then decodes discretely under the resulting context. A pretrained continuous drafter samples $z \sim p_\phi(z \mid c)$ in the representation space of a fixed encoder E. A

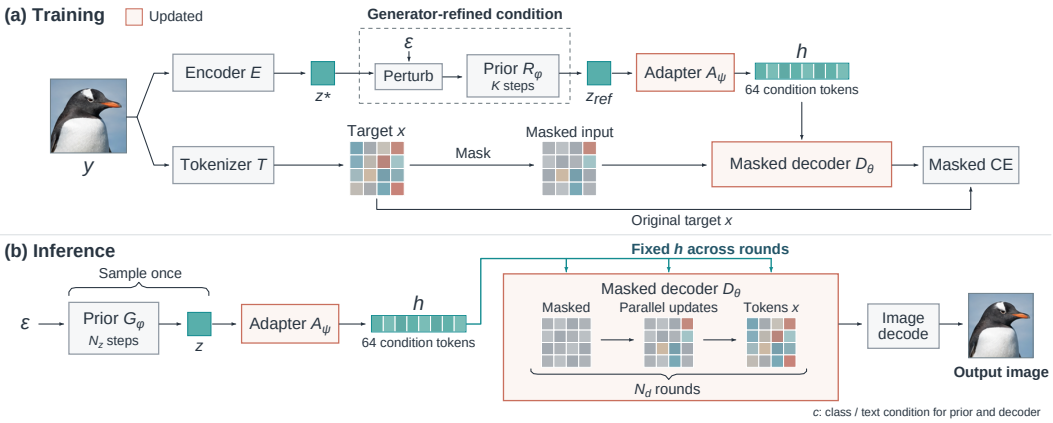

Figure 2: **Gentoo training and inference.** (a) During training, the prior refines a perturbed image representation in K steps. The adapter A_ψ and masked decoder D_θ learn to predict the original target tokens. (b) At inference, the prior samples z once; $h = A_\psi(z)$ stays fixed throughout decoding. The two stages have separate budgets N_z and N_d. Images and grids are schematic; baseline condition augmentation is omitted.

learned adapter maps the draft to context embeddings $h = A_\psi(z)$, which a pretrained masked decoder reads alongside the original condition c. Starting from a fully masked sequence, the decoder translates the draft into an initially incomplete sequence of discrete tokens, then progressively fills the remaining positions using both the draft and the tokens already committed. Translation and completion occur within the same decoding trajectory: the draft remains fixed while the visible set V_r grows. At round r, the remaining positions $M_r = [N] \setminus V_r$ are predicted in parallel,

$$q_{\theta,\psi}(\widetilde{x}_{M_r} \mid x_{V_r}, z, c) = \prod_{i \in M_r} q_{\theta,\psi,i}\big(\widetilde{x}_i \mid x_{V_r}, A_\psi(z), c\big), \tag{5}$$

and the sampler commits a block of these predictions. Writing $p^S_{\theta,\psi}(x \mid z, c)$ for the distribution induced by this decoding trajectory under sampler S, the complete generative model is

$$p^S_{\phi,\theta,\psi}(x \mid c) = \int \underbrace{p_\phi(z \mid c)}_{\text{draft continuously}} \underbrace{p^S_{\theta,\psi}(x \mid z, c)}_{\text{decode discretely}} dz. \tag{6}$$

The adapter parameters ψ and decoder parameters θ are adapted jointly, retaining the decoder's original vocabulary and sampling procedure, including guidance and confidence-based token selection.

Interface. The adapter, the only new module, projects the draft to 64 context tokens for decoder attention; only image-token predictions are used. During adaptation, a learned null context replaces these tokens with probability 0.1, enabling condition-free evaluation of the same adapted decoder.

Sampling budgets. The drafting budget N_z and decoding budget N_d are set independently. Counting one prior update or one decoding round as one network function evaluation (NFE), the total is $N_{\text{total}} = N_z + N_d$; translation and completion are both included in N_d. Allocating part of the budget to a shared draft aims to reduce the discrete rounds needed for compatible token commitments. We evaluate whether the saved decoding work outweighs the cost of drafting, reporting quality against total compute, time, and sampling steps. Compute and timing include both stages and their guidance branches.

Learning to use a generated draft. RCG conditions image generation on a compact global representation (Li et al., 2024a); Gentoo supplies a spatial draft through multiple context tokens. Sequential LADD also generates latents before tokens and supports pretrained encoders (Shariatian et al., 2025), but reports a latent-generation bottleneck for its continuous sequential variant in text experiments (Shariatian et al., 2025, Remark 5.1). Gentoo draws on independently pretrained representation generators (Zheng et al., 2026a; Tong et al., 2026) and adapts a pretrained masked decoder to their drafts. With the prior fixed, the training problem is to teach the decoder to translate generated drafts while preserving supervision from paired images and tokens.

3.3 DRAFTER-REFINED CONDITIONING

The pairing gap. For each training image y, the encoder and tokenizer provide a paired representation $z_\star = E(y)$ and token sequence $x = T(y)$. However, exact encodings may expose details that generated drafts do not reproduce, while independently sampled drafts lack corresponding target tokens. Drafter-refined conditioning transforms each paired encoding with the fixed drafter while retaining its original token targets.

Construction. We partially noise the paired encoding and evolve it with the drafter's velocity field. For a start time $\tau \sim \mathcal{U}[\tau_{\min}, \tau_{\max}]$ and noise $\epsilon \sim \mathcal{N}(0, I)$, we construct

$$z_\tau = (1-\tau)z_\star + \tau\epsilon, \qquad z_{\text{ref}} = R_\phi(z_\tau, \tau, c), \tag{7}$$

where R_ϕ integrates the velocity field over a descending time grid $t_0 = \tau > \cdots > t_K = t_{\text{end}}$:

$$u_0 = z_\tau, \qquad u_{k+1} = u_k + (t_{k+1} - t_k)\, v_\phi(u_k, t_k, c), \qquad z_{\text{ref}} = u_K. \tag{8}$$

The start time τ controls the balance between the paired encoding and noise, as in editing by partial noising (Meng et al., 2021). Intermediate start times retain information from the target encoding in the initialization while allowing the drafter to transform the condition. The decoder therefore learns from conditions transformed by the same dynamics used to generate drafts at inference.

Training objective. The decoder predicts the original tokens from the refined condition under the baseline's masked-token objective. Let $h_\psi(z_{\text{ref}}; \xi)$ denote the adapter together with the baseline's training-time condition augmentation, with randomness ξ. For a minibatch $\mathcal{B}$ with masks M_b,

$$\mathcal{L}(\theta, \psi) = \mathbb{E}\left[\frac{\sum_{b\in\mathcal{B}}\sum_{i\in M_b} \ell_{\text{base}}\big(q_{\theta,i}(\cdot \mid x_{\overline{M}_b}, h_\psi(z_{\text{ref},b}; \xi_b), c_b), x_{b,i}\big)}{\sum_{b\in\mathcal{B}} |M_b|}\right], \tag{9}$$

where ℓ_{base} is the baseline's masked cross-entropy. The adapter and decoder are updated jointly, while the drafter remains fixed. Refinement requires K gradient-free drafter evaluations per training example and adds no inference steps.

3.4 MODEL INSTANTIATIONS

We instantiate Gentoo with RAE and MaskBit for class-conditional ImageNet generation, and with Scale RAE and Meissonic for text-to-image generation. In both cases, the adapter supplies the continuous draft as context to the masked decoder while retaining the original class or text conditioning. Additional instantiations connect the same interface to Halton–MaskGIT and a text decoder (Appendices B.4 and E).

4 EXPERIMENTS

We examine how continuous drafts affect sequential coordination, generation quality, and total inference cost, and how refinement helps the decoder use these drafts.

4.1 EXPERIMENTAL SETUP

We combine MaskBit (Weber et al., 2024) with RAE (Zheng et al., 2026a) for ImageNet 256×256, and Meissonic (Bai et al., 2024) with Scale RAE (Tong et al., 2026) for text-to-image generation at 1024×1024. We report ImageNet FID and IS on 50,000 samples and text-to-image GenEval (Ghosh et al., 2023). The main tables compare Gentoo with each released discrete decoder and the prior's own image decoder, using 64-round discrete baselines as the budget references. Appendix A gives training and sampling settings and cost accounting for both stages.

4.2 CONTINUOUS CONTEXT SUPPORTS MORE PARALLEL DECODING

Quality becomes less dependent on sequential decisions. With a shared context, committing more tokens at once incurs a much smaller quality penalty (Figure 3). Reducing MaskBit from 64 to 16 rounds changes Gentoo's FID from 1.41 to 1.52, while the original decoder degrades substantially. Meissonic shows the same pattern: its quality depends strongly on the number of rounds

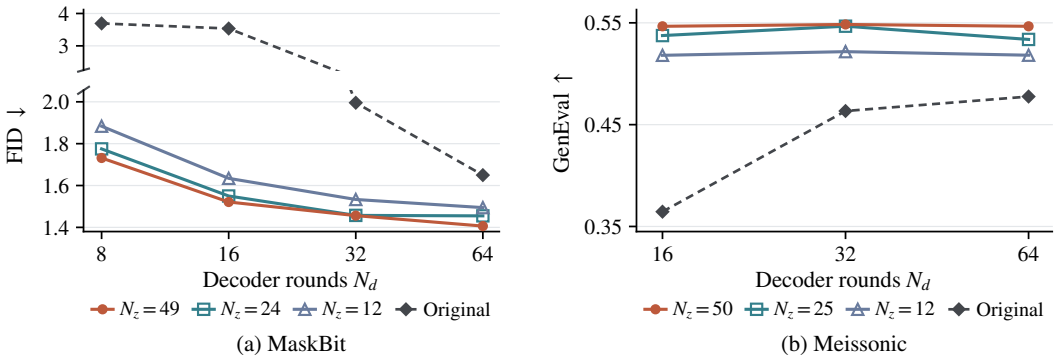

Figure 3: Inference-budget sensitivity. Colored curves fix prior steps N_z for one refined checkpoint; dashed curves show original decoders. (a) MaskBit FID with a broken axis. (b) Meissonic GenEval in fp32. Markers denote evaluated settings; lines connect them.

without context, whereas Gentoo's 16- and 64-round endpoint scores are nearly identical at every prior budget. Longer prior sampling improves GenEval even when additional discrete rounds contribute little. Together, these trends suggest that establishing the instance upstream reduces the coordination work left to progressive token revelation.

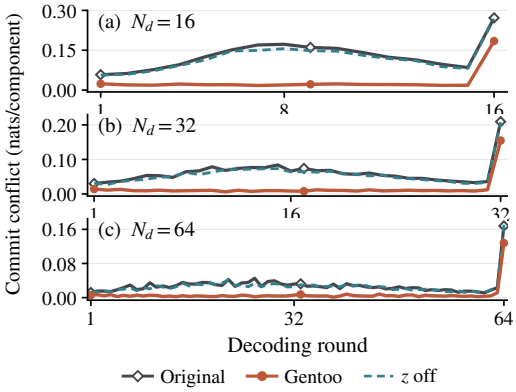

Figure 4: Commit conflict over MaskBit decoding (lower is better). Gentoo uses sampled z; "z off" disables context in the same decoder. Shading shows pointwise 95% intervals over 1,536 trajectories. Panel scales differ; all arms share the same guidance policy.

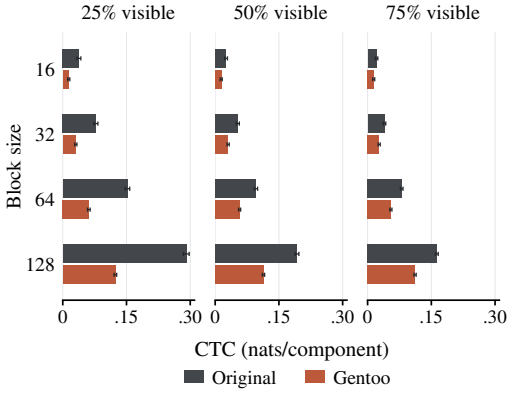

Figure 5: CTC (nats/component) on 1,024 validation images. Headers give visible token fractions. Gentoo uses matching encoded (oracle) z; whiskers show pointwise 95% intervals. Scores use unguided logits and average four chain orders.

Shared context reduces commit conflict. Figures 4 and 5 connect this robustness to more compatible token predictions on generated and held-out images. For the original decoder, it roughly halves each time the number of rounds doubles, from 0.152 at 16 rounds to 0.040 at 64, as expected when the dependence cost grows with block size. Gentoo lowers it by 54–63% at every budget, so that Gentoo at 16 rounds conflicts less than the original decoder at 32; disabling z in the same decoder returns conflict to the original level. On held-out images, matching encoded context also lowers CTC across all visible fractions and block sizes, and disabling context or swapping representations within the same class increases the gap in both cases. Appendix C gives the shared protocols, mathematical interpretations, and full diagnostic results.

4.3 Continuous coordination improves discrete generation

We next measure gains in generation quality and total inference cost. Supplementary budget and component analyses appear in Appendix B; comparisons with distilled, reward-tuned, and other generators appear in Appendix G.

Generation quality. At the higher-budget operating points, Gentoo lowers ImageNet FID from 1.65 to 1.41 and raises GenEval from 0.48 to 0.55 (Tables 1 and 2). All six GenEval categories improve over original Meissonic, with the largest gains in spatial relations and two-object composition. On ImageNet, higher recall at similar precision accompanies lower IS, reflecting broader coverage with less confident per-image class predictions. The best-FID ImageNet setting uses $2.32\times$ modeled compute relative to 64-round MaskBit.

Method	Prior steps	Decoder rounds	FID↓	IS↑
MaskBit (Weber et al., 2024)	–	256	1.52	<u>328.6</u>
MaskBit (Weber et al., 2024)	–	64	1.65	**332.5**
MaskBit (Weber et al., 2024)	–	16	3.81	278.8
RAE (Zheng et al., 2026a)	49	–	<u>1.51</u>	242.9
Gentoo–MaskBit ($z = 49$)	49	64	**1.41**	318.8
Gentoo–MaskBit ($z = 24$)	24	16	1.55	305.4

Table 1: ImageNet 256×256 (50,000 images). RAE reports published scores for its own decoder with the unguided 49-step prior. $z = k$ denotes prior evaluations in addition to decoder rounds; compute is not matched. **Bold**/<u>underline</u>: best/second-best, including ties; gray rows: Gentoo.

Method	Prior	Rounds	Single	Two	Count	Color	Position	Binding	Overall↑
Meissonic (Bai et al., 2024)	–	64	0.92	0.54	0.35	0.81	0.07	0.17	0.48
Meissonic (Bai et al., 2024)	–	16	0.79	0.38	0.17	0.70	0.05	0.09	0.36
Scale RAE (Tong et al., 2026)	50	–	0.95	0.59	<u>0.37</u>	0.63	<u>0.23</u>	0.15	0.49
Gentoo–Meissonic ($z = 50$)	50	16	**0.97**	**0.66**	**0.38**	**0.84**	<u>0.23</u>	**0.22**	**0.55**
Gentoo–Meissonic ($z = 25$)	25	16	<u>0.96</u>	<u>0.65</u>	0.34	<u>0.83</u>	**0.25**	<u>0.19</u>	<u>0.54</u>

Table 2: GenEval comparison (↑, fp32). Scale RAE uses its own 224^2 decoder on Gentoo's 50-step drafts; Meissonic and Gentoo output 1024^2 images.

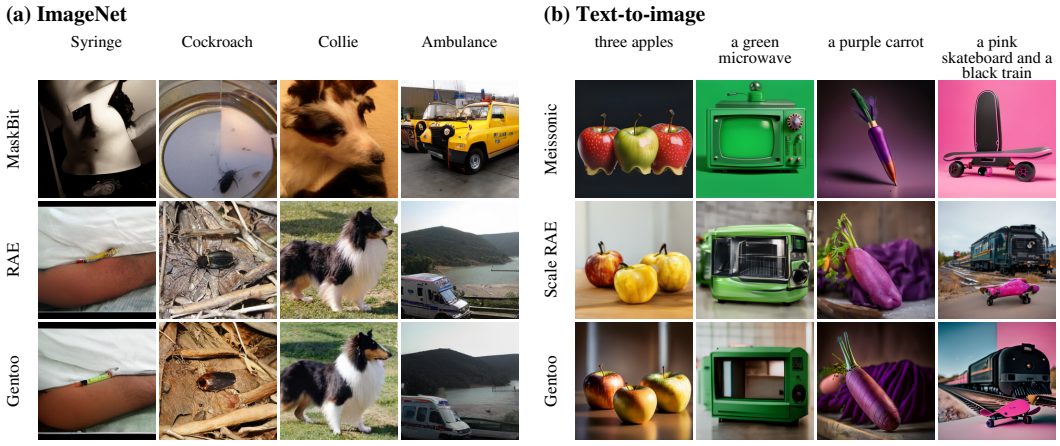

Figure 6: Selected samples. (a) Gentoo and RAE share drafts; ResNet-50 recognizes Gentoo's images but not RAE's, and MaskBit samples are misclassified. (b) Meissonic shares Gentoo's prompt and sample index; Scale RAE decodes Gentoo's draft at 224^2. Gentoo uses prior/decoder budgets 49/64 and 50/16; both original decoders use 64 rounds. Selection details: Appendix F.

Better quality at a lower total budget. Counting both stages against the original 64-round decoders, Gentoo uses 62.5% of the ImageNet NFE budget to improve FID from 1.65 to 1.55, with 16% lower measured inference time and $0.83\times$ modeled compute. On text-to-image generation, approximately 64% of the NFE budget improves fp32 GenEval from 0.48 to 0.54. The corresponding bf16 configuration improves GenEval from 0.495 to 0.542 while reducing estimated time from 1.411 to 0.669 s per image (53%). Both timings use an H800 at batch size four; text-to-image estimates sum prior, decoder, and language-conditioning times. Decoding savings cover drafting.

Complementary roles of drafting and completion. Figure 6 compares Gentoo with each component: RAE or Scale RAE decodes the same draft, while MaskBit or Meissonic supplies a class- or prompt-matched reference. Gentoo preserves the ImageNet drafts' composition while making the syringe and cockroach more recognizable. For the pink skateboard and black train, Scale RAE and Gentoo share the two-object layout, with Gentoo rendering the requested colors more clearly; Meissonic lacks a distinct train. Across GenEval categories, Gentoo with the 50-step prior matches or exceeds the stronger component in each category. Position accuracy matches Scale RAE's 0.23 versus Meissonic's 0.07; color accuracy reaches 0.84 versus Meissonic's 0.81 and Scale RAE's 0.63. Two-object composition and color binding exceed both components, supporting a draft that coordinates the scene and a discrete decoder that realizes its details and attributes.

4.4 Ablation studies

Training conditions originate from an image encoder, while inference conditions come from a prior. Table 3 tests how refinement bridges this gap by comparing direct and drafter-refined conditioning recipes under matched inference settings, retaining the original image tokens as supervision.

(a) ImageNet

FID↓			ADM IS↑			Target-class top-1 (%)↑		
w/o ref	with ref	Δ	w/o ref	with ref	Δ	w/o ref	with ref	Δ
1.487	**1.406**	-0.080	287.5	**305.8**	+18.4	88.47	**90.15**	+1.68

(b) GenEval

Variant	Overall	Single	Two	Counting	Colors	Position	Binding
w/o ref	0.509	**0.969**	0.606	0.325	0.790	0.225	0.138
with ref	**0.547**	0.966	**0.657**	**0.375**	**0.835**	**0.230**	**0.218**
Δ (pp)	+3.8	-0.3	+5.1	+5.0	+4.5	+0.5	+8.0

Table 3: Effect of drafter-refined training conditions. (a) ImageNet: FID, ADM IS, and target-class top-1 accuracy. (b) Overall and category-wise GenEval; Binding denotes color binding. Δ is with refinement minus without refinement, computed before rounding; accuracy and GenEval changes are in percentage points. Each pair uses the same inference settings.

Refinement improves how context is realized. The largest GenEval gains are in color binding, two-object composition, and counting, where the decoder must associate attributes and objects across the image. ImageNet gains class accuracy and IS with little change in precision and recall. Conditioning without refinement already provides comparable CTC and lower flip rates in the coordination diagnostics. These results distinguish the roles of shared context in coordinating token predictions and refinement in improving the realization of image content. Better generation under null and mismatched conditions further suggests that refinement improves the decoder's behavior when context is incomplete or unreliable.

Displacement-matched perturbations do not reproduce the refinement gain; continued finetuning alone does not reproduce Gentoo's gains (Appendix B.3).

5 Conclusion

Gentoo connects pretrained continuous priors and masked decoders through shared instance representations. It keeps generation in discrete tokens, offering a potential interface to unified multimodal models and diffusion language models and distinguishing it from latent diffusion models. The context reduces commit conflict, while generator-refined conditioning improves how the decoder uses generated representations. On MaskBit and Meissonic, Gentoo improves generation quality across decoding budgets, including configurations that outperform the original decoders at lower total inference cost. Despite these gains, Gentoo's applicability remains constrained by the availability of high-quality priors in the chosen representation space. Its efficiency advantage also depends on the sampling budget, as prior sampling overhead can offset the savings from fewer discrete decoding rounds, making the allocation between the two stages central to the quality–efficiency tradeoff.

Reproducibility Statement

The code is provided in the supplementary files. The appendix specifies adaptation settings, evaluation protocols, and inference cost accounting. The accompanying project records contain source metrics and deterministic scripts for the tables and figures. Quality results use one adaptation seed per configuration. The GenEval interval resamples evaluation prompts; commit intervals resample instances within three generation seeds; teacher-forced CTC intervals resample 512 class clusters of two held-out images each.

Statement of AI Use

Generative AI tools assisted with manuscript drafting, revision, and LaTeX formatting, as well as experiment implementation. The authors are responsible for reviewing AI-assisted content and verifying the final text, code, references, and empirical claims, and take full responsibility for the final manuscript.

References

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

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

Jinbin Bai, Tian Ye, Wei Chow, Enxin Song, Qing-Guo Chen, Xiangtai Li, Zhen Dong, Lei Zhu, and Shuicheng Yan. Meissonic: Revitalizing masked generative transformers for efficient high-resolution text-to-image synthesis. In *The Thirteenth International Conference on Learning Representations*, 2024.

Victor Besnier, Mickael Chen, David Hurych, Eduardo Valle, and Matthieu Cord. Halton scheduler for masked generative image transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83027–83047, 2025.

Michael Cardei, Huu Binh Ta, and Ferdinando Fioretto. Simple self-conditioning adaptation for masked diffusion models. *arXiv preprint arXiv:2604.26985*, 2026.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11305–11315. IEEE, 2022.

Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. SANA-Sprint: One-step diffusion with continuous-time consistency distillation. *arXiv preprint arXiv:2503.09641*, 2025.

Justin Deschenaux and Caglar Gulcehre. Beyond autoregression: Fast llms via self-distillation through time. In *International Conference on Learning Representations*, volume 2025, pp. 5007–5045, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. MDTv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Hongcan Guo, Qinyu Zhao, Yian Zhao, Shen Nie, Rui Zhu, Qiushan Guo, Feng Wang, Tao Yang, Hengshuang Zhao, Guoqiang Wei, et al. Continuous latent diffusion language model. *arXiv preprint arXiv:2605.06548*, 2026.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. *arXiv preprint arXiv:2412.04431*, 2024.

Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Distillation of discrete diffusion through dimensional correlations. *arXiv preprint arXiv:2410.08709*, 2024.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Dongwon Kim, Ju He, Qihang Yu, Chenglin Yang, Xiaohui Shen, Suha Kwak, and Liang-Chieh Chen. Democratizing text-to-image masked generative models with compact text-aware one-dimensional tokens. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18442–18452. IEEE, 2025.

Jean-Marie Lemercier, Tomas Geffner, Karsten Kreis, Morteza Mardani, Arash Vahdat, and Ante Jukić. Diladiff: Distilled latent-augmented diffusion for language modeling. *arXiv preprint arXiv:2605.23605*, 2026.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for end-to-end tuning with latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

José Lezama, Huiwen Chang, Lu Jiang, and Irfan Essa. Improved masked image generation with token-critic. In *European Conference on Computer Vision*, pp. 70–86. Springer, 2022.

Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2142–2152, 2023.

Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. *Advances in Neural Information Processing Systems*, 37: 125441–125468, 2024a.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-DiT: A powerful multi-resolution diffusion transformer with fine-grained Chinese understanding. *arXiv preprint arXiv:2405.08748*, 2024c.

Guotao Liang, Baoquan Zhang, Zhiyuan Wen, Zihao Han, and Yunming Ye. Improved masked image generation with knowledge-augmented token representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6817–6825, 2026.

Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete copula diffusion. In *International Conference on Learning Representations*, volume 2025, pp. 88953–88979, 2025.

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

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10609–10619. IEEE, 2022.

Patrick Pynadath, Jiaxin Shi, and Ruqi Zhang. Candi: Hybrid discrete-continuous diffusion models. *arXiv preprint arXiv:2510.22510*, 2025.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Beyond next-token: Next-x prediction for autoregressive visual generation. *arXiv preprint arXiv:2502.20388*, 2025.

Liran Ringel, Ameen Ali, and Yaniv Romano. Dependency-guided parallel decoding in discrete diffusion language models. *arXiv preprint arXiv:2604.02560*, 2026.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The diffusion duality. *Proceedings of machine learning research*, 267: 52584, 2025.

Alexander Shabalin, Viacheslav Meshchaninov, Egor Chimbulatov, Vladislav Lapikov, Roman Kim, Grigory Bartosh, Dmitry Molchanov, Sergey Markov, and Dmitry Vetrov. Tencdm: Understanding the properties of the diffusion model in the space of language model encodings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25110–25118, 2025.

Dario Shariatian, Alain Durmus, Umut Simsekli, and Stefano Peluchetti. Latent-augmented discrete diffusion models. *arXiv preprint arXiv:2510.18114*, 2025.

Junzhe Shen, Jieru Zhao, Ziwei He, and Zhouhan Lin. Codar: Continuous diffusion language models are more powerful than you think. *arXiv preprint arXiv:2603.02547*, 2026.

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

Qingyu Shi, Jinbin Bai, Zhuoran Zhao, Wenhao Chai, Kaidong Yu, Jianzong Wu, Yunhai Tong, Xiangtai Li, Xuelong Li, and Shuicheng Yan. Muddit: Liberating generation beyond text-to-image with a unified discrete diffusion model. In *International Conference on Learning Representations*, 2026.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Shengbang Tong, Boyang Zheng, Ziteng Wang, Bingda Tang, Nanye Ma, Ellis Brown, Jihan Yang, Rob Fergus, Yann LeCun, and Saining Xie. Scaling text-to-image diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2601.16208*, 2026.

Greg Ver Steeg and Aram Galstyan. Discovering structure in high-dimensional data through correlation explanation. *Advances in neural information processing systems*, 27, 2014.

Guanghan Wang, Yair Schiff, Subham Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *Advances in Neural Information Processing Systems*, 38:147282–147339, 2026a.

Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled diffusion transformer. *arXiv preprint arXiv:2504.05741*, 2025.

Yuqing Wang, Chuofan Ma, Zhijie Lin, Yao Teng, Lijun Yu, Shuai Wang, Jiaming Han, Jiashi Feng, Yi Jiang, and Xihui Liu. Cubic discrete diffusion: Discrete visual generation on high-dimensional representation tokens. *arXiv preprint arXiv:2603.19232*, 2026b.

Mark Weber, Lijun Yu, Qihang Yu, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. Maskbit: Embedding-free image generation via bit tokens. *arXiv preprint arXiv:2409.16211*, 2024.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. SANA: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations*, volume 2025, pp. 28240–28264, 2025.

Tianyu Xie, Shuchen Xue, Zijin Feng, Tianyang Hu, Jiacheng Sun, Zhenguo Li, and Cheng Zhang. Variational autoencoding discrete diffusion with enhanced dimensional correlations modeling. In *International Conference on Learning Representations*, volume 2026, pp. 39109–39132, 2026.

Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Keqi Wang, Yibin Wang, et al. Lumina-DiMOO: An omni diffusion large language model for multi-modal generation and understanding. *arXiv preprint arXiv:2510.06308*, 2025.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. In *International Conference on Learning Representations*, volume 2025, pp. 33769–33789, 2025.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Jaehoon Yoo, Wonjung Kim, Chanhyuk Lee, and Seunghoon Hong. Infinite mask diffusion for few-step distillation. *arXiv preprint arXiv:2605.10518*, 2026.

Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In *International Conference on Learning Representations*, volume 2024, pp. 765–783, 2024a.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024b.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

Guohui Zhang, Hu Yu, Xiaoxiao Ma, Yaning Pan, Hang Xu, Jie Huang, and Feng Zhao. Maskfocus: Focusing policy optimization on critical steps for masked image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5956–5966, 2026a.

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*, 2026b.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026a.

Haoyang Zheng, Xinyang Liu, 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*, volume 2026, pp. 29535–29571, 2026b.

Huangjie Zheng, Shansan Gong, Ruixiang Zhang, Tianrong Chen, Jiatao Gu, Mingyuan Zhou, Navdeep Jaitly, and Yizhe Zhang. Continuously augmented discrete diffusion model for categorical generative modeling. In *International Conference on Learning Representations*, volume 2026, pp. 34212–34242, 2026c.

Cai Zhou, Chenxiao Yang, Yi Hu, Chenyu Wang, Chubin Zhang, Muhan Zhang, Lester Mackey, Tommi Jaakkola, Stephen Bates, and Dinghuai Zhang. Coevolutionary continuous discrete diffusion: Make your diffusion language model a latent reasoner. *arXiv preprint arXiv:2510.03206*, 2025.

Yuanzhi Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. Di [m] o: Distilling masked diffusion models into one-step generator. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18606–18618. IEEE, 2025.

Yuanzhi Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. Soft-di [m] o: Improving one-step discrete image generation with soft embeddings. In *International Conference on Learning Representations*, volume 2026, pp. 2884–2931, 2026.

Le Zhuo, Ruoyi Du, Han Xiao, Yangguang Li, Dongyang Liu, Rongjie Huang, Wenze Liu, Lirui Zhao, Fu-Yun Wang, Zhanyu Ma, et al. Lumina-Next: Making Lumina-T2X stronger and faster with Next-DiT. *arXiv preprint arXiv:2406.18583*, 2024.

A Implementation and Evaluation

A.1 Adaptation settings

Table A1 gives the optimization settings for the two image instantiations. The tokenizer, representation encoder, and continuous prior are frozen throughout image adaptation.

Conditioning interfaces. For ImageNet, the frozen DINOv2 encoder of the RAE drafter supplies a $768 \times 16 \times 16$ representation, which a strided convolution turns into 64 condition tokens appended to MaskBit's 256 image tokens and its class token. For text-to-image generation, the frozen SigLIP2 encoder of the Scale RAE drafter supplies a $1152 \times 16 \times 16$ representation, and the adapter's 64 condition tokens are concatenated with the text hidden states before Meissonic's context projection, so that the text pathways of both the drafter and the decoder are retained. Both adaptation variants retain the baselines' condition dropout and token-space noise augmentation; refinement precedes the adapter.

	Gentoo–MaskBit	**Gentoo–Meissonic**
Training steps / evaluated step	15,000 / 8,000 (EMA)	15,000 / 15,000
Global batch size	1,024	64
Optimizer	AdamW	8-bit AdamW
Backbone / adapter learning rate	10^{-5} / 10^{-4}	10^{-4} / 10^{-4}
Adam β / weight decay	$(0.9, 0.96)$ / 0.045	$(0.9, 0.999)$ / 0.01
Gradient clipping	1	10
Label smoothing	0.1	0
Class or text dropout	0.1	0.1
Null / noisy / clean context	0.1 / 0.6 / 0.3	0.1 / 0.6 / 0.3
Training precision	fp32, TF32 enabled	bf16

Table A1: Image adaptation settings. Both objectives average categorical cross-entropy over masked targets. "Clean" denotes no additional token-space noise after representation construction.

MaskBit uses 4,000 warmup steps, a configured 100,000-step learning-rate horizon, a minimum learning-rate ratio of 0.1, and EMA decay 0.999. Its 256 spatial tokens each contain two 7-bit categorical components. Meissonic trains on 207,490 image–caption pairs from the same JourneyDB subset used by the continuation control, at 1024×1024, with eight images per device on eight devices. Its caption representations are cached. Both variants add variance-preserving noise to projected condition tokens in the noisy branch, with a level drawn uniformly from $[0.05, 0.2]$.

Representation normalization and integration. MaskBit uses the normalized DINOv2-B representation expected by RAE. Refinement uses bf16 and six Euler steps, with a grid uniform in raw time before the prior's time shift $\sqrt{48}$ and its prescribed nonzero numerical endpoint.

Prior-sampled states use this same conversion at inference for both adapted variants. The two recipes share the adapter architecture, target tokens, and subsequent condition augmentation; the Meissonic ablation compares these complete conditioning recipes. Meissonic integration uses time shift $\sqrt{72}$ and endpoint 0.0084222. In both image priors, the start time is uniform on $[0.4, 0.9]$ in model time.

A.2 Image evaluation

ImageNet quality evaluations use 50,000 images and generation seed 0, except for the explicitly labeled 10,000-image condition controls. The local FID and IS evaluator is cleanfid's legacy-TensorFlow InceptionV3W, with ADM reference statistics for FID and ten splits for IS. Additional ImageNet readouts use the ADM suite's IS, sFID, precision, and recall, together with target-class accuracy.

The main Meissonic evaluations use the official GenEval scorer on 553 prompts, four images per prompt, with generation seed 42. The main comparisons use fp32; the bf16 evaluations use the same benchmark. Paired score-difference intervals use 2,000 prompt-bootstrap draws, retaining all images belonging to each sampled prompt and recomputing the aggregate score. These intervals describe prompt-sampling variation at fixed checkpoints. The supplementary SDXL position com-

parison and qualitative selection follow the official GenEval protocol with a Hugging Face port of the Mask2Former scorer; those scores are reported within that shared implementation.

Sampling settings. MaskBit uses CFG 7.1 (10.5 for Gentoo at eight rounds), with guidance power 3 for the main original reference and 1 for Gentoo; the commit probe uses power 1 for all arms. The round-sensitivity curves select the better recorded original-decoder guidance policy at each budget. Meissonic uses CFG 9/12 for the original/Gentoo models, and CFG 9 for every arm of the continuation comparison. Sampling settings are fixed within each refinement pair.

The main RAE reference reports the published unguided result. Our 49-step reproduction, with guidance scale 1 and the released pixel decoder, gives FID 1.50 and local IS 255.8 on the same 50,000-image protocol. The Scale RAE reference decodes 50-step prior samples with its released 224×224 pixel decoder and uses the same 553-prompt GenEval protocol.

A.3 Inference cost

For ImageNet, one conditioned decoder branch costs 1.27 original decoder forwards and one prior step costs 2.75. The 64-round original sampler uses two guidance branches per round, giving

$$C_{\mathrm{IN}}(N_z, N_d) = \frac{2(1.27)N_d + 2.75N_z}{128}. \tag{10}$$

This accounts for inference through the prior and the longer conditioned decoder sequence. Training and distillation are separate costs.

ImageNet wall-clock timings use an H800 at batch size four in fp32 with TF32 enabled, averaging five repetitions after two warmup runs. The timed interval includes prior sampling, the condition adapter, discrete sampling, tokenizer decoding, and conversion to CPU uint8 output; it excludes model loading and file writing. The original 64-round MaskBit takes 0.2083 s per image, compared with 0.1754 s for Gentoo with 24 prior evaluations and 16 decoder rounds, a 15.8% reduction.

Meissonic component timings are measured on an H800 at batch size four, with the prior and decoder in bf16. Per-image decoder time is 1.4109 s for Meissonic at 64 rounds and 0.4115 s for Gentoo at 16 rounds. Prior sampling takes 0.1041, 0.2249, and 0.4596 s for 12, 25, and 50 steps. Language-model conditioning uses fp32 and takes 0.13 s per prompt, or 0.0325 s per image when shared by four images. For the quality–cost curves, we sum these components and normalize by the original decoder time; this is a component sum rather than an end-to-end service measurement.

Direct online measurements on A100. Table A2 additionally measures both complete generation paths on an A100-SXM4-80GB, at batch sizes one and four. ImageNet uses fp32 with TF32 enabled; text-to-image generators use bf16, with Gentoo's language model in fp32. Models remain resident, and each measurement averages five synchronized repetitions after two warmup runs. The online interval includes fresh text conditioning when applicable, fresh prior sampling, discrete and pixel decoding, and transfer to CPU image arrays; loading and file writing are excluded. A text-to-image batch shares one prompt's conditioning. FLOPs count matrix products, convolutions, and attention arithmetic, counting multiplication and addition separately and excluding sampler bookkeeping and most elementwise operations. These operator counts are separate from the relative-cost model in Equation 10.

At batch four, the measured online paths reduce ImageNet time from 0.3314 to 0.3223 s per image and text-to-image time from 2.5577 to 1.2207 s per image. ImageNet is slower at batch one, showing that the time benefit depends on batching as well as hardware and sampling budget.

Model	Prior / decoder	Time (s/image) Batch 1	Time (s/image) Batch 4	TFLOPs/image Batch 1	TFLOPs/image Batch 4
MaskBit	0 / 64	0.6821	0.3314	20.97	20.97
Gentoo–MaskBit	24 / 16	0.7740	0.3223	14.50	14.50
Meissonic	0 / 64	3.5935	2.5577	197.21	197.14
Gentoo–Meissonic	25 / 16	1.8730	1.2207	90.26	88.74

Table A2: Measured online inference cost on A100. ImageNet outputs are 256^2 and text-to-image outputs are 1024^2. Prior and decoder columns give the respective update counts; time and FLOPs include the full measured path defined above.

B Additional Image-Generation Experiments

B.1 Quality and total inference budgets

Figure A1 extends the round-sensitivity analysis to quality versus the combined cost of continuous drafting and discrete decoding. Each point includes both stages, using the ImageNet compute model or Meissonic component timings. The paired training variants show how refinement shifts this tradeoff across prior and decoder budgets.

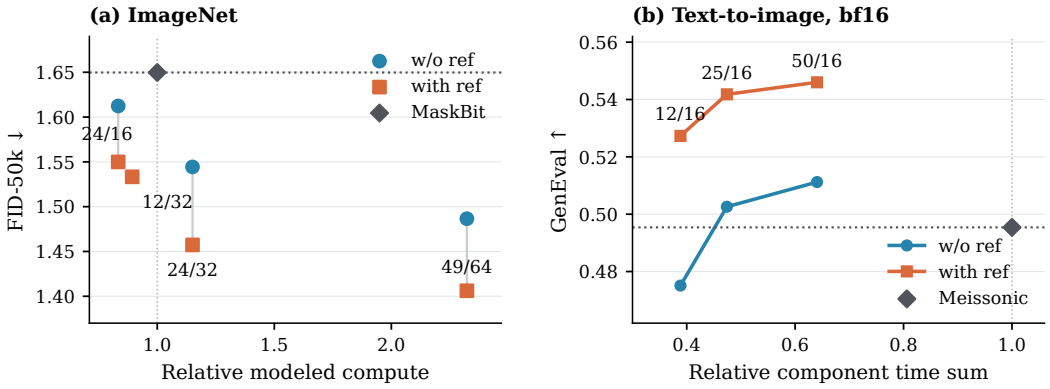

Figure A1: Refinement across inference-cost settings. Blue: w/o ref; orange: with ref. Labels give prior evaluations / discrete rounds. ImageNet uses FID-50k and modeled compute; Meissonic uses bf16 GenEval and component timings. Both cost definitions are given in Appendix A.3. The gray point marks the original decoder.

B.2 Component behavior across decoder budgets

ADM evaluation. Table A3 reports additional ImageNet readouts across decoder budgets, including precision, recall, and target-class accuracy. We use the ADM suite's TorchScript Inception graph, shuffled 5,000-image splits for ADM IS, and improved precision and recall with $k = 3$ against 10,000 reference images. Target-class top-1 is measured by an independent ImageNet ResNet-50. The Inception conditional entropy is $H_c = \mathbb{E}_x H[y \mid x]$; the marginal label entropy lies between 6.898 and 6.903 nats over these rows.

Coverage and class confidence. Relative to original MaskBit at 64 rounds, Gentoo with 49 prior evaluations and 64 decoder rounds improves recall at similar precision, indicating broader coverage with approximately maintained fidelity. Its lower IS accompanies less confident per-image class predictions, while the marginal class entropy stays nearly unchanged. Refinement recovers part of this class confidence while retaining the coverage gain. Compared with RAE's own pixel decoder on samples from the same unguided prior, Gentoo improves FID, sFID, and target-class accuracy.

Method	DLM NFE	FID↓	sFID↓	ADM IS↑	Prec.↑	Rec.↑	Top-1 (%)↑	H_c
RAE, own pixel decoder	–	1.505	5.298	243.8	0.789	0.637	82.43	1.342
MaskBit	64	1.650	4.955	317.6	0.806	0.602	91.83	1.076
Gentoo ($z = 49$), w/o ref	64	1.487	4.691	287.5	0.802	0.613	88.47	1.177
Gentoo ($z = 49$), with ref	64	1.407	4.615	305.8	0.802	0.617	90.15	1.115
Gentoo ($z = 24$), w/o ref	32	1.545	4.789	278.7	0.801	0.614	87.86	1.210
Gentoo ($z = 24$), with ref	32	1.458	4.724	297.8	0.806	0.615	89.46	1.142
Gentoo ($z = 24$), w/o ref	16	1.612	5.003	277.3	0.797	0.614	87.75	1.214
Gentoo ($z = 24$), with ref	16	1.550	4.985	292.0	0.798	0.615	89.05	1.162
Gentoo ($z = 49$), with ref	8	1.731	5.515	301.4	0.789	0.613	89.94	1.128
MaskBit	32	1.995	5.905	300.4	0.794	0.608	91.12	1.130
MaskBit	16	3.804	8.776	265.9	0.762	0.604	89.23	1.252

Table A3: ADM-suite results on 50,000 ImageNet images per setting. All FID values here use the ADM evaluator; ADM IS is distinct from the local IS in the main ImageNet comparison. Top-1 is a percentage and H_c is in nats. The RAE row decodes the unguided 49-step samples of the same prior with RAE's own pixel decoder. Gentoo denotes Gentoo–MaskBit; gray rows use refinement.

B.3 Refinement and adaptation controls

These controls distinguish refinement from further decoder training and random perturbations, then test the adapted decoders under altered inference conditions. The coordination diagnostics complement these generation results: direct conditioning already lowers commit flips and gives CTC comparable to refined conditioning.

B.3.1 Matched training displacement

We compare direct encoder conditions, generator-refined conditions, and a random-direction perturbation in the same representation space on ImageNet. At each sampled start time, its magnitude follows the measured refinement displacement profile, with a fixed calibration factor to match the average drift. The remaining augmentation, optimizer, data order, and seed are shared, and all three arms use their step-8,000 EMA checkpoint. Mean relative displacement is 0.28275 for the control and 0.28271 for refinement; mean cosine similarity to the initial encoding is 0.9603 and 0.9567, respectively. Figure A2(a) shows that the magnitude-matched perturbation gives little FID change at $49/64$ and worsens FID at $24/16$.

Refinement across budgets and precision. Panel (b) complements the ImageNet displacement control with Meissonic refinement pairs at 16 discrete rounds. Refinement improves bf16 GenEval at all three prior budgets. Increasing prior steps from 12 to 50 raises refined GenEval from 0.5273 to 0.5460 in bf16 and from 0.5181 to 0.5466 in fp32.

B.3.2 Continued finetuning

We continue training the original decoders on the adaptation data without adding instance context. MaskBit runs for 8,000 updates with global batch size 1,024 and its backbone adaptation optimizer; evaluation uses the EMA checkpoint. Meissonic runs for 15,000 updates with global batch size 64, learning rate 10^{-4}, and masked-only cross-entropy, updating the original transformer while freezing the VQ tokenizer and text encoder.

Figure A3 compares the completed runs at 64 discrete rounds. MaskBit's FID changes from 1.650 to 1.642; local IS is 331.0 and target-class top-1 is 91.72% after continuation. Meissonic's GenEval changes from 0.4776 to 0.4644. At 16 rounds with the same sampling settings, continuation raises Meissonic from 0.3645 to 0.3983. Additional decoder training alone does not reproduce the gains from continuous conditioning.

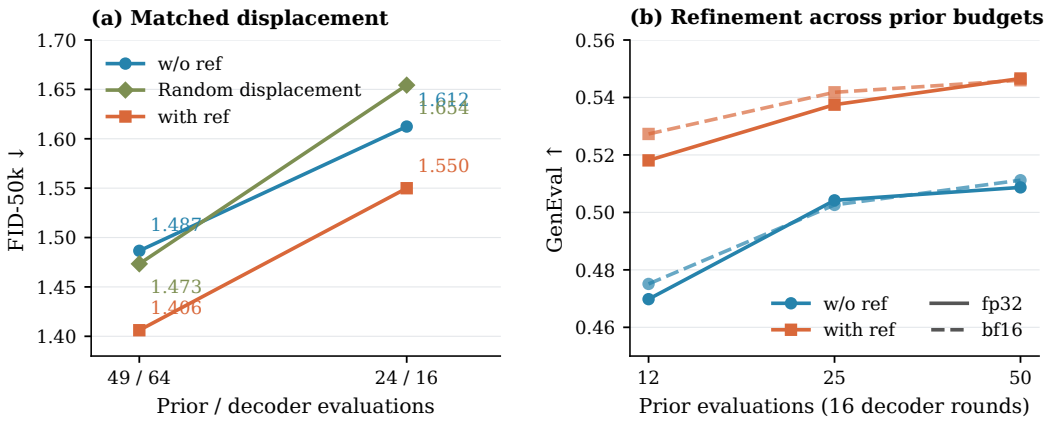

Figure A2: Training-condition controls. (a) Equal-displacement adaptation on ImageNet, evaluated on 50,000 images at two inference budgets. (b) Meissonic refinement at 16 discrete rounds across prior budgets and evaluation precisions; each setting uses 2,212 images.

Shared decoder guidance. The Meissonic arms share a CFG scale of 9 and 64 discrete rounds. GenEval increases from 0.4776 for the released decoder to 0.5149 for adaptation with direct encoder conditions and 0.5409 with refined conditions; continued finetuning gives 0.4644. Thus, the conditioning gains persist at the original decoder's guidance scale. Both conditioned variants use 50 prior evaluations in addition to the shared decoder budget.

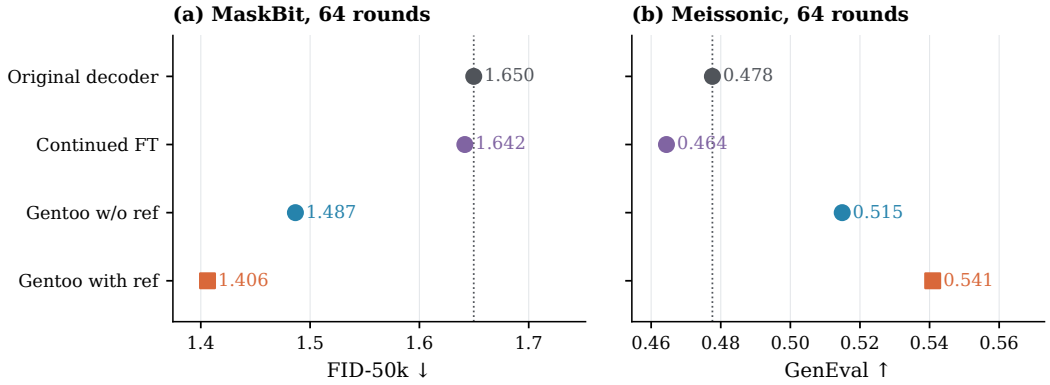

Figure A3: Continued finetuning controls. ImageNet uses 50,000 images; Meissonic uses 553 prompts with four images each in fp32, with shared sampling settings across all arms (Appendix A.2). Gentoo uses 49 prior evaluations for MaskBit and 50 for Meissonic.

B.3.3 Altered inference context

Figure A4 holds the adapted checkpoint fixed while changing the inference condition. Null context uses the learned null token; the Meissonic mismatch uses a prior sample from another prompt. The refinement gain in GenEval is 3.70 percentage points with null context (95% paired interval $[1.46, 5.89]$) and 7.86 points with mismatched context ($[5.38, 10.71]$).

B.4 Transfer to Halton–MaskGIT

We additionally connect the RAE drafter to the Halton–MaskGIT decoder (Besnier et al., 2025). Gentoo–Halton uses 49 prior evaluations followed by 32 discrete rounds, reaching FID 3.09 compared with 5.16 for the released Halton decoder (Table A4). This configuration uses encoder-derived training conditions and the step-2,000 EMA adaptation checkpoint.

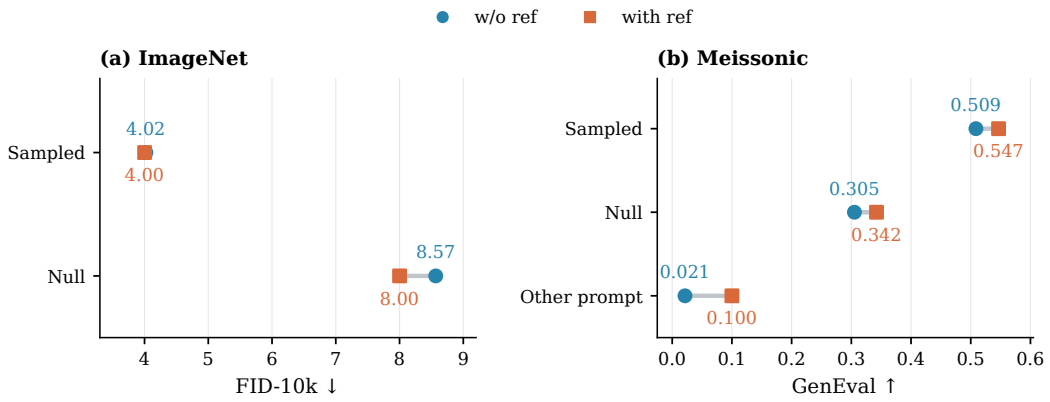

Figure A4: Inference-condition controls for the two adapted variants. ImageNet uses the serial 10,000-image protocol at $49/64$. Meissonic uses fp32 at $50/16$; sampled and null conditions use 2,212 images, while the other-prompt condition uses one image for each of 553 prompts. Segments pair the two training variants within each condition; points are completed-run estimates.

Method	**DLM NFE**	**FID↓**	**IS↑**
Halton–MaskGIT (Besnier et al., 2025)	32	5.1622	285.9
Gentoo–Halton ($z = 49$)	32	3.0875	270.8

Table A4: Halton backbone transfer on 50,000 ImageNet images. Both decoders use 32 rounds; Gentoo–Halton additionally uses the unguided 49-step RAE prior and is adapted without training-time refinement. FID and IS use the same local ImageNet evaluator.

C Additional Decoder Diagnostics

C.1 Commit conflict: protocol, interpretation, flips, and entropy

For each round t, the sampler's confidence ranking gives the newly committed block B_t. Independent Bernoulli draws with probability $1/2$ partition it into subsets A_t and C_t. Starting from the same pre-commit canvas, we reveal the sampled values in A_t and rescore those in C_t, then exchange the subsets. Let $p_{t,i}$ and $p_{t,i}^{+S}$ denote the guided categorical probabilities before and after revealing the sibling subset S. With $S(i)$ denoting the opposite half, the per-sample conflict is

$$C = \frac{\sum_t \sum_{i \in B_t} \left[\log p_{t,i}(\hat{x}_i) - \log p_{t,i}^{+S(i)}(\hat{x}_i) \right]}{\sum_t |B_t|}. \tag{11}$$

Positive values indicate that revealing sibling commitments lowers support for the sampled token. Negative values are retained. To interpret the expectation, consider exact conditionals and a fixed block. Given the context, the value $\hat{x}_i$ is drawn from $p(X_i \mid \cdot)$ independently of the sibling values $\hat{x}_S$, so averaging over $\hat{x}_i$ gives

$$\mathbb{E}_{\hat{x}_i}[\log p(\hat{x}_i \mid \cdot) - \log p(\hat{x}_i \mid \cdot, \hat{x}_S)] = \mathrm{KL}\big(p(X_i \mid \cdot) \,\big\|\, p(X_i \mid \cdot, \hat{x}_S)\big) \geq 0, \tag{12}$$

which vanishes for all sibling values exactly when X_i is conditionally independent of X_S. Expected conflict is therefore nonnegative, and it is positive when tokens committed in the same round depend on each other. With a learned decoder, guidance, and confidence-based selection, the measured quantity is a model-based probe of this effect. The flip rate replaces the log-probability difference by an indicator that the most probable token changes. We also retain the pre-reveal entropy.

The probe uses a separate random-number generator and leaves the sampled trajectory unchanged. We use 512 instances per generation seed for three seeds, sampling 512 classes without replacement within each seed. The same 49-step prior samples are reused across the adapted arms and

16-, 32-, and 64-round budgets. All arms use CFG 7.1 with guidance power 1, keeping the round-specific guidance weight fixed during each sibling reveal. The measurement covers 511 committed components per image, excluding the last component filled by the final returned prediction. Means weight committed components equally. Intervals use 2,000 paired whole-instance bootstrap resamples within generation-seed strata at fixed checkpoints.

Figure A5 adds flip rate and pre-reveal entropy to the conflict curves in the main paper. Matching context reduces flips across the trajectory, while the null-context arm moves toward the original decoder. Low flip rates without refinement are consistent with coordination gains before refinement.

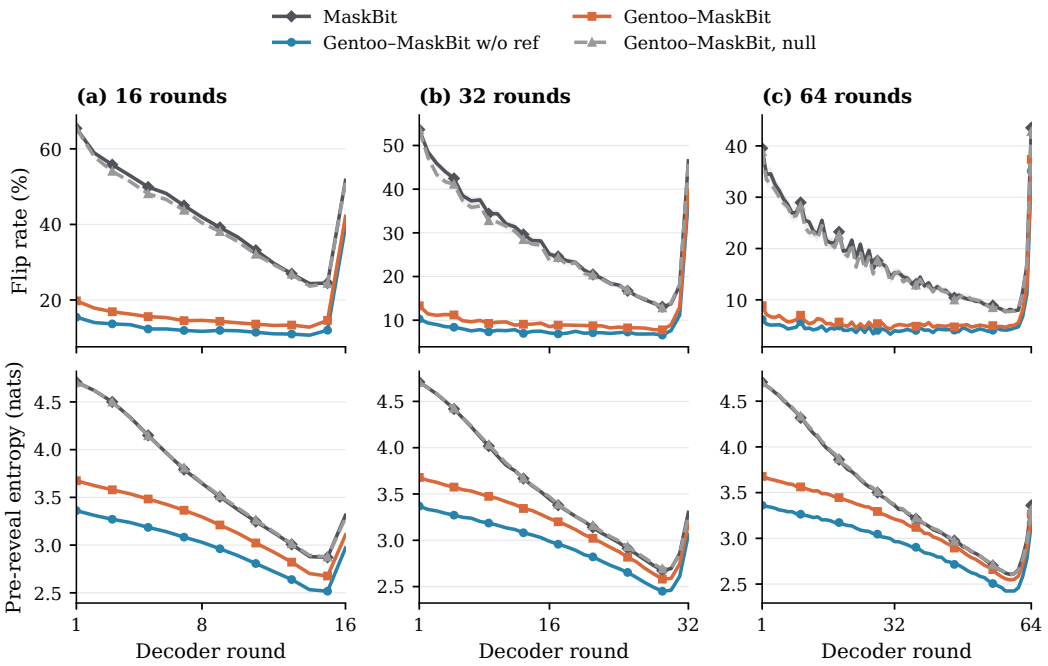

Figure A5: Commit flip rate and pre-reveal entropy over decoding rounds. All four arms use the same guidance policy, with 1,536 generated instances per arm and budget. Shading gives pointwise 95% bootstrap intervals. Vertical scales vary across columns.

C.2 Parallel and chain likelihoods

We use two held-out validation images from each of 512 sampled classes, with deterministic 256×256 center crops. The tokenizer provides target components x^*, and the frozen encoder provides matching oracle representations z for all 1,024 images.

At each prespecified visible fraction ρ, we sample the visible positions V uniformly, then uniformly sample a block B of K masked components. We evaluate the parallel marginal score and a teacher-forced chain score for each of four random orders π:

$$\ell_{\text{par}} = \sum_{i \in B} \log p_\theta(x_i^* \mid x_V^*, c, z), \tag{13}$$

$$\ell_{\text{chain}}^{\pi} = \sum_{j=1}^{K} \log p_\theta(x_{\pi_j}^* \mid x_V^*, x_{\pi_{<j}}^*, c, z), \tag{14}$$

$$\text{CTC} = \frac{1}{K}\left(\frac{1}{4}\sum_{\pi} \ell_{\text{chain}}^{\pi} - \ell_{\text{par}}\right). \tag{15}$$

The released model has no latent condition. Disabling Gentoo's condition uses its learned null/drop condition. CTC is the per-component gap between parallel and mean chain NLL, averaging orders in log space. Scores use unguided class-conditional logits, retaining negative estimates.

The grid uses $\rho \in \{0.25, 0.50, 0.75\}$ and $K \in \{16, 32, 64, 128\}$, with four random chain orders per image and setting. The original-versus-refined CTC comparison appears in Figure 5. At $\rho = 0.5, K = 32$, the same-class shuffle exchanges the two encoded conditions within each class, preserving the class but changing the instance match (Table A5). The pointwise intervals use 2,000 paired bootstrap draws of the 512 class clusters, retaining both images and the shuffled pair. The identity below separates the contributions of data dependence and model error to the expected CTC.

Method / condition	CTC	Parallel NLL	Chain NLL
MaskBit	0.0532	3.2100	3.1569
Gentoo–MaskBit, w/o ref	0.0306	2.5545	2.5239
Gentoo–MaskBit, with ref	0.0305	2.6554	2.6250
Gentoo–MaskBit, null	0.0536	3.2586	3.2051
Gentoo–MaskBit, shuffled	0.0587	4.1706	4.1119

Table A5: Instance-match controls at 50% visible context and block size 32. All readouts are nats per categorical component. "Shuffled" exchanges representations within the same class.

C.3 Parallel–chain diagnostic identity

Write q_i for the decoder probabilities in (15), and let $U = (X_V, C)$ without instance context or $U = (X_V, Z, C)$ with it. For a fixed block B of size K and an ordering π, define

$$q_{\mathrm{par}}(x_B \mid U) = \prod_{i \in B} q_i(x_i \mid U),$$
$$q^{\pi}_{\mathrm{chain}}(x_B \mid U) = \prod_{j=1}^{K} q_{\pi_j}(x_{\pi_j} \mid U, x_{\pi_{<j}}). \tag{16}$$

Each fixed-order chain product is a normalized joint distribution, even when the decoder's conditionals across masks need not agree with one common joint distribution. Let $\widehat{\mathrm{CTC}}_q$ denote the per-block statistic in (15). For finite divergences, its expectation under the data distribution is

$$K\,\mathbb{E}_p[\widehat{\mathrm{CTC}}_q] = \mathrm{TC}_p(X_B \mid U) + \varepsilon_{\mathrm{par}} - \varepsilon_{\mathrm{chain}}, \tag{17}$$

where

$$\varepsilon_{\mathrm{par}} = \sum_{i \in B} \mathbb{E}_U\, \mathrm{KL}(p(X_i \mid U) \,\|\, q_i(X_i \mid U))\,,$$
$$\varepsilon_{\mathrm{chain}} = \mathbb{E}_{\pi,U}\, \mathrm{KL}(p(X_B \mid U) \,\|\, q^{\pi}_{\mathrm{chain}}(X_B \mid U))\,. \tag{18}$$

The order average is over the four probe orders. The result follows by writing each expected NLL as entropy plus KL divergence and using $\mathbb{E}_U \mathrm{KL}(p(X_B \mid U) \| q_{\mathrm{par}}(X_B \mid U)) = \mathrm{TC}_p(X_B \mid U) + \varepsilon_{\mathrm{par}}$.

Thus the expected CTC combines conditional total correlation and decoder approximation errors. It equals conditional total correlation per component when both sets of conditionals match the data.

D Proofs

Proof of Proposition 1. The conditional total-correlation explanation identity (Ver Steeg & Galstyan, 2014) follows by subtracting (2) with and without Z as context:

$$\begin{aligned}\mathrm{TC}(X_B \mid X_V, C) - \mathrm{TC}(X_B \mid X_V, Z, C) &= \sum_{i \in B} \big[\mathrm{H}(X_i \mid X_V, C) - \mathrm{H}(X_i \mid X_V, Z, C)\big] \\ &\quad - \big[\mathrm{H}(X_B \mid X_V, C) - \mathrm{H}(X_B \mid X_V, Z, C)\big] \\ &= \sum_{i \in B} \mathrm{I}(X_i; Z \mid X_V, C) - \mathrm{I}(X_B; Z \mid X_V, C),\end{aligned}$$

since each of the $|B| + 1$ entropy differences is a conditional mutual information by definition. ∎

Proof of Corollary 1. Sum (3) over r with $B = B_r$ and $V = V_r$. Because the blocks partition $[N]$ and $V_r = B_1 \cup \cdots \cup B_{r-1}$ collects the earlier ones, the chain rule of mutual information gives $\sum_{r=1}^{R} \mathrm{I}(X_{B_r}; Z \mid X_{V_r}, C) = \mathrm{I}(X; Z \mid C)$, which yields (4). For sequential decoding each block is one position, so both sides of (3) vanish each round; for a single round, $V_1 = \emptyset$ and $B_1 = [N]$. ■

E ADDITIONAL TEXT-GENERATION EXPERIMENTS

On OpenWebText, we examine how sampled representations affect quality and coverage within the same trained text decoder. With temperatures selected for comparable entropy, Gentoo achieves lower Gen-PPL than its null-context counterpart, together with higher feature-space recall and Vendi diversity and lower Fréchet distance to held-out text. The gain in distributional coverage at similar token entropy suggests that shared context supports sequence variation beyond sampling temperature. The following sections provide training and evaluation settings, the full temperature sweep, condition controls, and overlap and repetition analyses.

E.1 REPRESENTATION, REFINEMENT, AND DISTILLATION

We initialize the discrete backbone from the released small MDLM checkpoint (Sahoo et al., 2024), using the GPT-2 tokenizer and 1,024-token training blocks. A frozen Qwen3-Embedding-0.6B encoder (Zhang et al., 2025) provides contextual token features. We pool these into 64 equal windows of the GPT-2 sequence, aligning the two tokenizers by byte offsets, and retain 32 whitened PCA coordinates fitted on training data. Thus, $z \in \mathbb{R}^{64\times 32}$ describes the whole sequence. A linear projection, layer normalization, and learned slot and null vectors map z to 64 prefix tokens of width 768.

The continuous prior is a separately trained 38.8M-parameter rectified flow on these text representations. It has eight transformer blocks of width 512, is trained for 60,000 updates with batch size 256 and learning rate 10^{-4}, and is then frozen. Refinement uses its exponential moving average (EMA) weights. In standardized coordinates, the training condition starts at $u_{t_s} = (1 - t_s)\epsilon + t_s z$, where $\epsilon \sim \mathcal{N}(0, I)$ and $t_s \sim \mathcal{U}[0.4, 0.9]$. Six Euler steps integrate the frozen velocity field to $t = 0.999$, after which we undo the standardization. Here $t = 0$ denotes noise and $t = 1$ denotes data, reversing the image convention through $t_{\text{text}} = 1 - t_{\text{image}}$.

We first adapt MDLM and the prefix adapter for 20,000 updates with refined conditions, batch size 512, and learning rate 3×10^{-4}. The adapted EMA weights initialize self-distillation through time (SDTT) (Deschenaux & Gulcehre, 2025). Distillation uses the same refined condition and null-condition mask for teacher and student, the backward-KL objective, batch size 128, and learning rate 6×10^{-5} for 80,000 updates. The decoding budget is halved every 10,000 updates from 1,024 to a minimum of eight steps. Condition dropout is 0.1 in both stages. The representation encoder and flow remain frozen throughout adaptation and distillation. At inference, the flow draws z from Gaussian noise with 50 Euler steps, and the student decodes under this fixed sample. The $T = 8$ sampler uses eight diffusion updates followed by the default final argmax denoising pass. Inference also includes the continuous prior and the prefix tokens.

E.2 EVALUATION PROTOCOL

Gen-PPL uses GPT-2-large and the MDLM evaluation code. H_{seq} is the mean within-sequence entropy of the empirical GPT-2 token histogram, in nats. Distributional features are masked-mean pooled GPT-2-large final hidden states over the first 512 tokens, with dimension 1,280. FD is their Fréchet distance to 1,000 held-out reference features. For the nearest-neighbor precision proxy P_{NN}, a generated feature is counted if its nearest reference feature is within that reference point's fifth-neighbor radius. Exchanging the generated and reference sets defines recall R_{NN}. Vendi (Friedman & Dieng, 2022) is the exponential spectral entropy of K/n, where K is the cosine-similarity matrix of the n features. Real-text calibration gives FD 23.36, $P_{\text{NN}} = 0.447$, $R_{\text{NN}} = 0.480$, and Vendi 16.00; FD and neighbor metrics use two disjoint real-text draws, and Vendi uses the reference draw.

The temperature and condition experiments use the step-80,000 non-EMA student checkpoint, with 1,000 sequences per arm from eight generation seeds, 125 sequences each. Decoding uses a 1,024-token canvas, eight diffusion updates plus a final argmax pass, and no nucleus truncation. Each experiment shares decoding seeds and the sampled latent cache across its applicable arms. Null context is the learned null prefix in the same trained student.

E.3 Full temperature sweep

Figure A6 reports all six metrics at $\tau \in \{0.95, 1.00, 1.05\}$, using the same reference features across arms. The entropy-matched comparison uses sampled context at $\tau = 0.95$ and null context at $\tau = 1.05$ ($H_{\text{seq}} = 4.929$ and 4.921). At each fixed temperature, sampled context improves FD, P_{NN}, R_{NN}, and Vendi, while also raising Gen-PPL and within-sequence entropy.

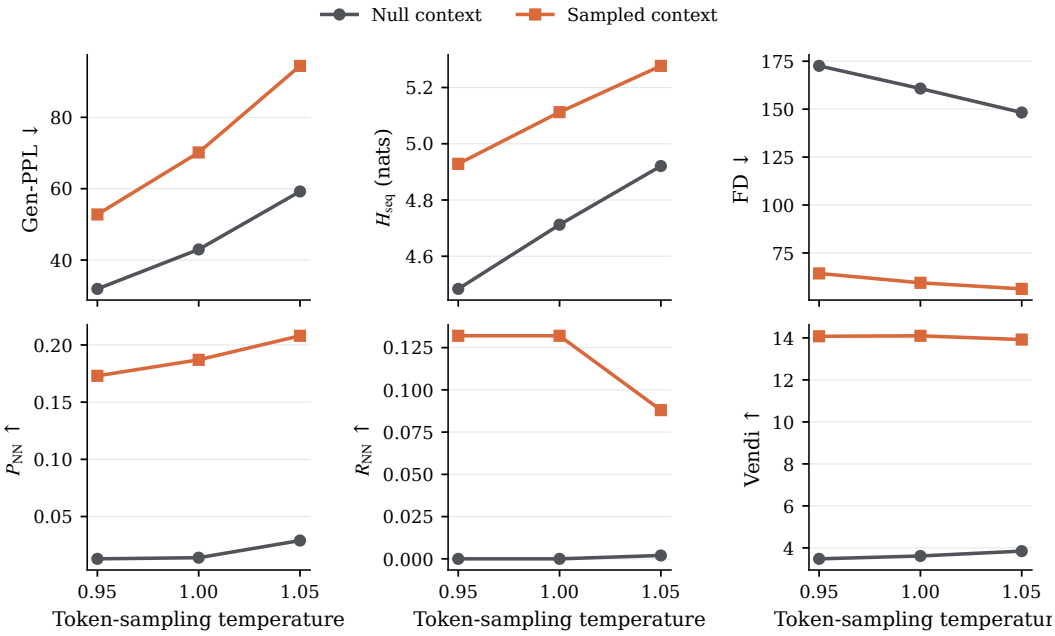

Figure A6: Full text temperature sweep with 1,000 sequences per setting. Orange: sampled context; gray: null context. All settings share the trained decoder, feature reference, and evaluation protocol. Lines connect the three evaluated temperatures.

E.4 Inference-condition controls

Figure A7 uses a separate 1,000-sequence cohort per arm to compare null context across seven temperatures, Gaussian and uniform latent replacements, shuffled slots of a prior sample, and the unmodified prior sample. All non-null arms use $\tau = 1$. Across the tested null temperatures, R_{NN} stays at or below 0.003 and Vendi at or below 4.86. Gaussian and uniform replacements also have zero measured recall. Sampled and slot-shuffled conditions retain higher recall (0.117 and 0.127) and Vendi (14.13 and 13.82), with Gen-PPL 69.38 and 79.16. These readouts separate the effect of the sampled representation from simply adding a latent array or raising temperature.

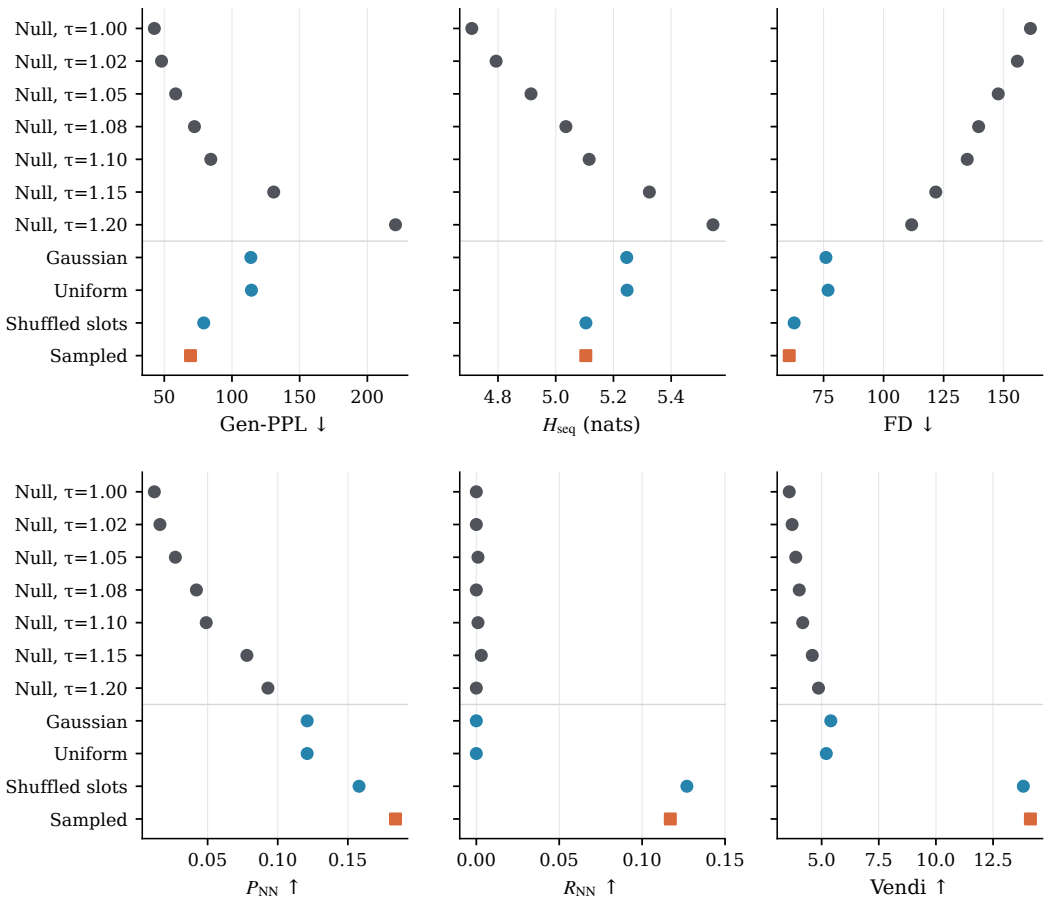

Figure A7: Text condition controls at the same trained checkpoint. Gray circles: null context with varying temperature; blue circles: replacement or shuffled latent conditions; orange squares: sampled context. Each point summarizes 1,000 sequences. The temperature sweep uses a separate cohort.

Feature-window sensitivity. Table A6 recomputes neighbor and diversity metrics on three fixed windows of these outputs at temperature 1. The reference uses a separate 1,000-sequence held-out draw at the same token positions. Sampled context has higher recall and Vendi throughout.

Token positions	P_{NN} ↑		R_{NN} ↑		Vendi↑	
	Null	Sampled	Null	Sampled	Null	Sampled
1–256	0.074	0.293	0.002	0.177	4.13	17.04
1–512	0.034	0.176	0.000	0.143	3.58	14.13
513–1024	0.097	0.207	0.000	0.119	3.48	14.22

Table A6: Fixed-window text metrics at temperature 1. Token positions are one-indexed.

E.5 Text overlap and repetition

Table A7 supplements feature-space coverage with an eight-token overlap check on 1,000 sequences per arm at temperature 1. Training overlap is the fraction of generated eight-grams found in a fixed random subset of 300,000 OpenWebText training blocks. Cross-sample duplication counts eight-gram occurrences shared by at least two output sequences; within-sample repetition averages the fraction of non-unique eight-grams in each sequence. Sampled context has fewer cross-sample

duplicates than null context, alongside higher within-sample repetition and training overlap. The held-out text row calibrates these statistics on natural text; the comparison concerns the specified eight-gram and training-subset coverage.

Context / reference	Training overlap	Cross-sample duplicate	Within-sample repeat
Held-out text	2.3856	0.3636	1.3946
Null context	0.1570	0.1119	0.0229
Sampled context	0.2223	0.0322	0.0880

Table A7: Eight-gram overlap and repetition, in percent. All rows contain 1,000 sequences tokenized with GPT-2 and truncated to at most 1,024 tokens.

F Additional Qualitative Examples

F.1 Selection of the main component comparisons

Figure 6 uses automatic screening followed by visual selection. For text-to-image, Meissonic (64 rounds) and Gentoo ($z = 50$, 16 rounds) are regenerated with seed $42+$ prompt index, the original cached drafts, and fp32, and Scale RAE decodes the same cached drafts with its own pixel decoder; qualifying samples are those that the GenEval implementation of Appendix A.2 marks correct for Gentoo and incorrect for both Meissonic and Scale RAE. For ImageNet, we take the 40 classes with the largest per-class gap in ResNet-50 top-1 accuracy between Gentoo and RAE, sample four RAE drafts per class, and decode each with RAE's pixel decoder and with Gentoo; qualifying drafts are those whose Gentoo image the classifier recognizes and whose RAE decoding it does not. On these 160 drafts, the classifier recognizes 120 Gentoo images and 102 RAE decodings. MaskBit shows the first misclassified sample of each class in the main 50,000-image sample set.

F.2 Layout following against VAE latent diffusion

We compare Gentoo–Meissonic with SDXL (Podell et al., 2024), a VAE latent diffusion model, on the 100 GenEval position prompts. SDXL base 1.0 generates four 1024×1024 samples per prompt with 50 steps and guidance 5.0, seeded like our GenEval runs. Both models follow the official GenEval protocol using the same scorer as the qualitative selection, under which Gentoo reaches a position accuracy of 0.240 and SDXL 0.098. These scores use the recorded 400-image sets. On matched prompts and sample indices, Gentoo is correct and SDXL incorrect in 83 cases, and the reverse holds in 26.

Selection and regeneration. Figure A8 uses the lowest-indexed prompt for each relation with at least three correct Gentoo samples and no correct SDXL samples. Gentoo samples are regenerated with the original seeds and cached drafts; 42 of the 48 regenerated samples reproduce the recorded GenEval outcomes, and borders mark the verdicts for the displayed frames.

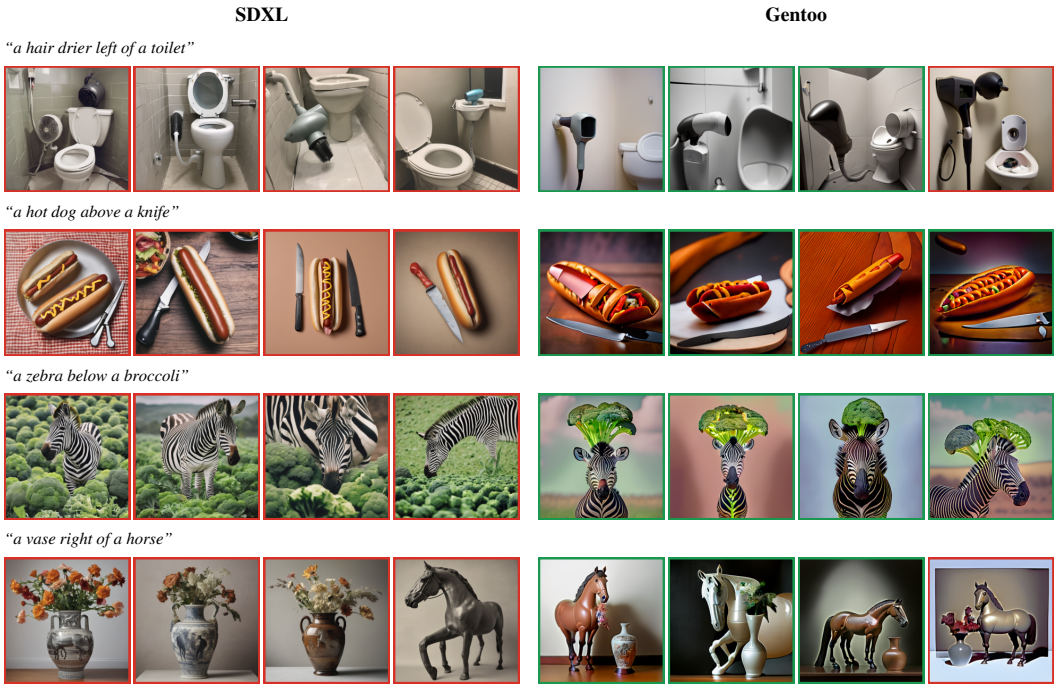

Figure A8: GenEval position prompts: SDXL (left) and Gentoo–Meissonic (right), four samples each. Green and red borders mark correct and incorrect GenEval verdicts. In these selected examples, SDXL merges objects or misses the requested relation, while Gentoo depicts the specified arrangement.

G External Generator Comparisons

G.1 Few decoder rounds and one-step students

Table A8 places Gentoo's few-round settings beside published one-step students of the same decoders. Di[M]O (Zhu et al., 2025) distills MaskBit and Meissonic into one round at FID 2.89 and GenEval 0.43. Soft-Di[M]O (Zhu et al., 2026) adds an adversarial loss on ImageNet or CLIP and ImageReward finetuning for text-to-image generation, reaching FID 1.56 and GenEval 0.53.

Gentoo's image decoders are adapted without decoder distillation, using the same guidance settings as the main comparisons. On ImageNet, eight decoder rounds with 24 or 49 prior evaluations reach FID 1.78 and 1.73; on GenEval, 16 decoder rounds with 25 prior evaluations reach 0.54. Gentoo supplies a continuous draft to condition token generation, while the one-step students learn a shorter decoding trajectory. Our text extension also combines conditioning with an eight-step self-distilled decoder.

(a) ImageNet 256×256

Method	Prior steps	Decoder rounds	FID↓	IS↑
MaskBit (Weber et al., 2024)	–	16	3.81	278.8
Di[M]O–MaskBit (Zhu et al., 2026)	–	1	2.89	310.1
Soft-Di[M]O–MaskBit (Zhu et al., 2026)	–	1	1.56	273.2
Gentoo–MaskBit ($z = 24$)	24	8	1.78	310.8
Gentoo–MaskBit ($z = 49$)	49	8	1.73	315.0
Gentoo–MaskBit ($z = 24$)	24	16	1.55	305.4

(b) GenEval

Method	Prior steps	Decoder rounds	GenEval↑
Meissonic (Bai et al., 2024)	–	16	0.36
Di[M]O–Meissonic (Zhu et al., 2025)	–	1	0.43
Soft-Di[M]O–Meissonic (Zhu et al., 2026)	–	1	0.53
Gentoo–Meissonic ($z = 25$)	25	16	0.54
Gentoo–Meissonic ($z = 50$)	50	16	0.55

Table A8: Few decoder rounds without decoder distillation, beside published one-step students of the same decoders. (a) ImageNet 256×256 with 50,000 samples; (b) GenEval in fp32. One-step rows report published results; Gentoo rows are evaluated here and shown in gray. Prior evaluations and decoder rounds are listed separately.

G.2 Published results of other generators

Table A9 provides reference results from other generator families. External rows use published scores; the autoregressive and VAE-based ImageNet rows use the re-evaluation in RAE (Zheng et al., 2026a). The 64-round Meissonic row is our reproduction, and Gentoo rows are evaluated in this work. The reported configurations differ in training data, model size, sampling, and evaluation implementation.

For ImageNet, guidance follows each source: Gentoo guides the discrete decoder, while the AutoGuidance RAE row also evaluates a weaker prior model at each step (Karras et al., 2024). On GenEval, MaskFocus uses GenEval and CLIP reward post-training, and Infinity-2B uses prompt rewriting; these are marked in the table. Steps lists each cited sampling budget, with Gentoo's prior and decoder steps shown separately; n/r denotes settings not reported in the source table.

(a) ImageNet 256×256**, 50,000 samples** **Method**	**Guidance**	**FID↓**	**IS↑**
Autoregressive			
VAR-d30 (Tian et al., 2024)	CFG	1.73	350.2
MAR-H (Li et al., 2024b)	CFG	1.55	303.7
xAR-H (Ren et al., 2025)	CFG	1.24	301.6
Latent diffusion with a VAE			
DiT-XL/2 (Peebles & Xie, 2023)	CFG	2.27	278.2
SiT-XL/2 (Ma et al., 2024)	CFG	2.06	270.3
MDTv2-XL/2 (Gao et al., 2023)	CFG	1.58	314.7
LightningDiT-XL/2 with VA-VAE (Yao et al., 2025)	CFG	1.35	295.3
REPA (SiT-XL/2) (Yu et al., 2025)	CFG	1.29	306.3
DDT-XL/2 (Wang et al., 2025)	CFG	1.26	310.6
REPA-E (SiT-XL/2) (Leng et al., 2025)	CFG	1.15	304.0
Latent diffusion with a representation autoencoder			
RAE, DiT^{DH}-XL (Zheng et al., 2026a)	none	1.51	242.9
RAE, DiT^{DH}-XL (Zheng et al., 2026a)	AutoGuidance (Karras et al., 2024)	1.13	262.6
Masked discrete diffusion			
MaskGIT (Chang et al., 2022)	none	6.18	182.1
TiTok-S-128 (Yu et al., 2024b)	CFG	1.97	281.8
MAGVIT-v2 (Yu et al., 2024a)	CFG	1.78	319.4
CubiD-XXL (Wang et al., 2026b)	CFG	1.88	247.0
MaskBit, 256 rounds (Weber et al., 2024)	CFG	1.52	328.6
Di[M]O–MaskBit, 1 round (Zhu et al., 2026)	CFG	2.89	310.1
Soft-Di[M]O–MaskBit, 1 round (Zhu et al., 2026)	CFG	1.56	273.2
Gentoo–MaskBit ($z = 49$, 64 rounds)	CFG (decoder)	1.41	318.8
Gentoo–MaskBit ($z = 24$, 16 rounds)	CFG (decoder)	1.55	305.4

(b) GenEval **Method**	**Res.**	**Params (B)**	**Steps**	**GenEval↑**
Masked or unified discrete generators				
Show-o (Xie et al., 2025)	n/r	1.3	n/r	0.68
MaskGen-L (VQ) (Kim et al., 2025)	256	0.57	n/r	0.53
MaskGen-XL (VQ) (Kim et al., 2025)	256	1.1	n/r	0.57
Muddit (Shi et al., 2026)	512	1.0	64	0.61
MMaDA-8B (Yang et al., 2025)	512	8.0	50	0.63
Lumina-DiMOO (Xin et al., 2025)	1024	8.0	64	0.88
Meissonic, 64 rounds (Bai et al., 2024)	1024	1.0	64	0.48
Di[M]O–Meissonic (Zhu et al., 2025)	1024	1.0	1	0.43
Soft-Di[M]O–Meissonic (Zhu et al., 2026)	1024	1.0	1	0.53
MaskFocus† (Zhang et al., 2026a)	1024	1.0	64	0.76
Gentoo–Meissonic ($z = 25$, 16 rounds)	1024	1.0 + 2.4	25 + 16	0.54
Gentoo–Meissonic ($z = 50$, 16 rounds)	1024	1.0 + 2.4	50 + 16	0.55
Continuous diffusion				
SDXL (Podell et al., 2024)	1024	2.6	n/r	0.55
Lumina-Next (Zhuo et al., 2024)	1024	2.0	n/r	0.46
SD3-Medium (Esser et al., 2024)	1024	2.0	n/r	0.62
Hunyuan-DiT (Li et al., 2024c)	1024	1.5	n/r	0.63
SANA-1.6B (Xie et al., 2024)	1024	1.6	n/r	0.66
SANA-Sprint-1.6B (Chen et al., 2025)	1024	1.6	1	0.74
Autoregressive				
Infinity-2B‡ (Han et al., 2024)	1024	2.0	n/r	0.73

Table A9: Reference results on (a) ImageNet and (b) GenEval, with sources described above. †GenEval-reward post-training. ‡Prompt rewriting.

G.3 MEASURED COST OF RELEASED GENERATORS

Table A10 reports locally measured inference cost for released generators under the A100 online protocol of Appendix A.3. ImageNet models use fp32 with TF32, and text-to-image generators use bf16. DiT uses DDPM sampling; SiT uses SDE Euler sampling on a 250-point grid; VAR uses ten autoregressive scales; MAR uses 256 outer iterations with 100 inner diffusion steps and linear guidance. The text-to-image runs use released Diffusers checkpoints: SANA-1.6B, SANA-Sprint-1.6B, Lumina-Next-SFT, and Hunyuan-DiT-v1.2, with the settings listed in the table. These runs measure cost only; the published quality scores above retain their source configurations.

			Time (s/image)		TFLOPs/image
Model	Sampler setting	Guidance	Batch 1	Batch 4	Batch 4
ImageNet 256×256					
DiT-XL/2	250 steps	1.5	3.2701	2.0185	119.24
SiT-XL/2	250 grid points	1.8	6.4304	3.9297	237.87
VAR-d20	10 scales	4.0	0.1268	0.0458	1.54
MAR-L	256×100	3.0	71.3455	19.2058	118.15
Text-to-image 1024×1024					
SANA-1.6B	20 steps	4.5	1.2895	1.1041	127.87
SANA-Sprint-1.6B	1 step	4.5	0.1904	0.1502	11.86
Lumina-Next-SFT	30 steps	4.0	14.7276	14.0554	953.85
Hunyuan-DiT-v1.2	50 steps	5.0	16.0117	15.0080	1447.38

Table A10: Online inference cost of additional released generators on A100. Guidance is the configured sampler scale; MAR's setting gives outer iterations $\times$ inner diffusion steps. Timing includes text conditioning where applicable and image decoding through CPU output.