REFINE OR PRESERVE? ACCEPTANCE-GATED CORRECTION FOR AUTOREGRESSIVE IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-hoc refinement can repair degraded autoregressive image samples, but the same intervention can damage outputs that are already well tuned. We therefore ask a prior question: should a generated sample be refined at all? We introduce Acceptance-Gated Refinement (AGR), a two-level correction policy for frozen autoregressive image generators. An image-level *router* chooses between exact preservation and refinement, while a token-level *gate* proposes masked-token replacements and commits only edits that pass a refiner-score test. The router is derived from the gate's own initial refiner pass and calibrated per domain using tuned and severely degraded validation samples; no degradation label is required at test time. Across FFHQ, CelebA-HQ, and ImageNet, AGR bypasses at least 98.5% of tuned samples, with observed FID changes of only $+0.06$, $+0.08$, and -0.01, while reducing severe-input FID from 102.71 to 47.36, 81.26 to 30.17, and 96.25 to 58.09. Mild inputs excluded from router fitting also improve. A router fitted only on temperature shifts improves unseen 4-bit quantized samples while bypassing top-k and top-p truncated samples for which forced refinement is harmful. At matched edit rates, random editing degrades mild inputs whereas score-gated editing improves them. Across thirteen settings in three domains, refinement gain increases with the input's score deficit under the refiner. These results show that a refiner can provide not only candidate corrections, but also a practical signal for deciding *when to refine and when to preserve*.

1 INTRODUCTION

Autoregressive (AR) image generators produce discrete tokens sequentially, typically committing to each choice without revisiting it (Esser et al., 2021; Lee et al., 2022; Sun et al., 2024). A separate bidirectional refiner can reconsider a completed sequence and repair local inconsistencies. Yet post-hoc refinement creates an asymmetric risk: the same intervention that improves a degraded sample can damage an output that was already well tuned. A higher refiner score does not by itself guarantee a better generated image. The resulting problem is therefore not simply how to refine, but first whether to refine or to preserve.

When regeneration with better generator settings is available, it is generally preferable to repairing a finished sample. We instead study settings in which completed token sequences are available but regeneration is impractical or unavailable – for example, archived generations of mixed provenance, outputs produced by a quantized deployment model whose original full-precision weights are unavailable, or heterogeneous collections assembled from fixed generation pipelines. The goal is *selective post-hoc correction*: preserve plausible samples exactly while intervening on inputs for which refinement is likely to help.

We introduce **Acceptance-Gated Refinement (AGR)**, which separates this decision at two levels. At the *image level*, a validation-calibrated router chooses between exact bypass and refinement. Unflagged samples are returned token-for-token unchanged. At the *token level*, a frozen bidirectional refiner ranks uncertain positions, proposes masked-token replacements, and accepts only proposals that pass a score test. The router is obtained from the gate's own initial refiner pass, so it requires no additional model. Both the autoregressive generator and the trained refiner remain fixed during

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.

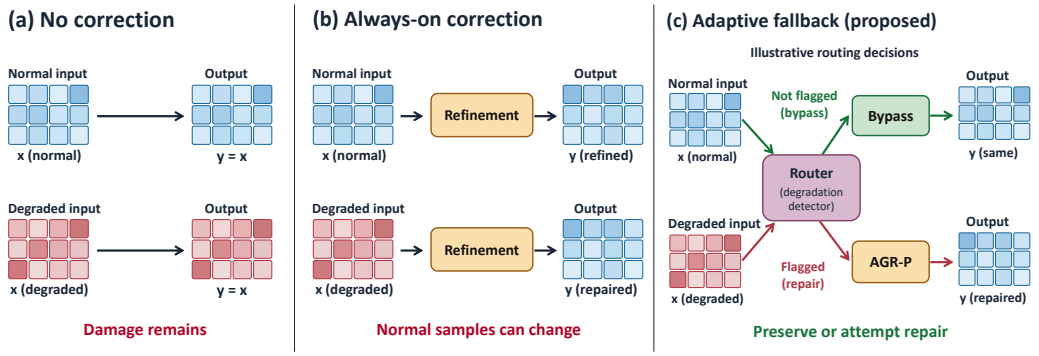

Figure 1: **Selective intervention for heterogeneous inputs.** No correction preserves samples but leaves degradation untreated; always-on correction can unnecessarily change normal inputs. AGR routes between exact bypass and gated correction. Token grids and routing decisions are schematic illustrations, not experimental image pairs.

inference, and AGR requires no test-time knowledge of temperature, quantization level, truncation rule, or other degradation label.

The two decisions are linked by a common empirical signal. Inputs whose refiner scores fall below those of well-tuned samples are also those for which gated refinement tends to provide the largest gain. Across thirteen settings in three domains, refinement benefit grows with this refiner-score deficit. A router calibrated only from tuned and severely temperature-degraded validation samples consequently generalizes to conditions excluded from fitting: it opens on unseen 4-bit weight quantization, where correction improves FID, while remaining closed on top-k and top-p truncation, where forced refinement is harmful. The router is therefore not simply a detector for the degradation used during calibration; it estimates whether the refiner has useful corrective leverage on the input.

Experiments on FFHQ, CelebA-HQ, and ImageNet support both sides of this decision. On tuned inputs, AGR bypasses at least 98.5% of samples and changes aggregate FID by only $+0.06$, $+0.08$, and -0.01. On severe inputs, it reduces FID from 102.71 to 47.36 on FFHQ, from 81.26 to 30.17 on CelebA-HQ, and from 96.25 to 58.09 on ImageNet. Mild inputs, which are excluded from router fitting, also improve. Rate-matched random-edit controls show that the gain is not explained merely by changing fewer tokens: random editing worsens mild FFHQ and CelebA-HQ, whereas score-gated editing improves both. Unlabeled mixtures further show why a single always-on refinement strength cannot simultaneously protect tuned samples and repair degraded ones.

Our contributions are: **(i) the router**: an image-level rule read from the gate's own first step that chooses between exact bypass and gated correction, fitted once per domain without test corruption labels; **(ii) the gate**: acceptance-gated token correction, with ungated and rate-matched random controls showing that the acceptance criterion, not the edit budget, produces the repair; and **(iii) a measured account of preservation and repair** across three image domains, unlabeled mixtures, five generator backbones, and five intervention settings excluded from fitting, together with the empirical rule that ties them together: the gate's gain grows with the input's score deficit, and the router thresholds that deficit.

2 Related Work

2.1 Autoregressive image generation and token revision

Two-stage AR generators model sequences from a learned visual tokenizer (van den Oord et al., 2017; Esser et al., 2021; Lee et al., 2022; Sun et al., 2024). VAR and MAR change the generation representation or prediction order (Tian et al., 2024; Li et al., 2024); RandAR enables flexible generation orders (Pang et al., 2025). MaskGIT, Token-Critic, and MUSE use bidirectional models for iterative generation or token selection (Chang et al., 2022; Lezama et al., 2022; Chang et al., 2023). Our setting instead starts with a completed sequence from a frozen generator. We use masked generation as a proposal mechanism and study when its post-hoc edits should be applied.

TensorAR introduces overlapping next-tensor predictions and discrete noising for token revision (Cheng et al., 2025). BIGFix trains with random-token injection to enable correction during generation (Besnier et al., 2025). These methods make revision part of the generation procedure and require retraining or fine-tuning the generator, so they are not comparable in our setting, where the generator is frozen and only its samples are available; we cite them as the closest generator-side alternatives. Our contribution is an external operating policy: a separately trained, tokenizer-compatible refiner routes and edits samples without updating the generator during refiner training or inference. VSR provides another refinement comparison (Wang et al., 2025); we report the class-conditioned ImageNet control alongside the face results.

2.2 Inference-time selection and acceptance

Several inference-time methods select or accept outputs with an auxiliary model. Image-level rejection sampling uses an auxiliary evaluator to select samples (Azadi et al., 2018; Turner et al., 2019), while verifier-based test-time selection allocates computation among candidate solutions (Cobbe et al., 2021; Snell et al., 2024; Brown et al., 2024). Our policy selects both images to edit and token proposals to retain. Unlike lossless speculative decoding (Leviathan et al., 2023; Chen et al., 2023), the gate is not designed to preserve a target sampling distribution. LANTERN relaxes acceptance to accelerate visual decoding (Jang et al., 2025); our composition test (Appendix G) concerns downstream repair and does not establish a quality-matched speedup.

2.3 Refiner scores as quality proxies

Refiner scores have known limits as quality measures. Pseudo-log-likelihood (PLL) sums masked conditional scores (Besag, 1975; Salazar et al., 2020). Such scores can support token selection without defining a compatible joint distribution. We distinguish masked conditionals in sequential gating from the gate's unmasked scores, and evaluate both against image quality. Neither the routing statistic nor the acceptance score is an independent quality oracle.

3 Acceptance-Gated Refinement

Given a completed token sequence from a frozen generator, AGR first decides whether to intervene, then decides which token edits to retain. In this section we fix notation (Section 3.1), describe the token-level gate (Section 3.2), specify the image-level router with exact bypass and its calibration (Section 3.3), and relate AGR to other operating policies (Section 3.4). The full procedure does not require a test image's temperature, quantization level, or corruption label.

3.1 Preliminaries

A frozen AR generator produces a token sequence $\mathbf{x} = (x_1, \ldots, x_T) \in \mathcal{V}^T$ over a codebook $\mathcal{V}$; it is never updated and is queried only to draw $\mathbf{x}$. A separate bidirectional refiner f_θ is trained by masked-token denoising on sequences from the same tokenizer and image domain, and is likewise frozen at inference. For an unmasked input $\mathbf{x}$, define the refiner's output distribution and current-token score as

$$r_i(v; \mathbf{x}) = \mathrm{softmax}(f_\theta(\mathbf{x})_{[i]})_v, \qquad \hat{s}_i(\mathbf{x}) = \log r_i(x_i; \mathbf{x}). \tag{1}$$

The softmax is over codebook entries, excluding the mask symbol. In contrast to a masked conditional, this score is computed with x_i visible to the refiner. Class-conditional refiners also receive the generation label, omitted here for readability.

3.2 The token-level gate

Each iteration uses three refiner forwards (Figure 2). **Rank:** evaluate $f_\theta(\mathbf{x})$ and select the $k_s = \max(1, \lceil \rho_s T \rceil)$ positions with smallest maximum probability $c_i = \max_v r_i(v; \mathbf{x})$. **Propose:** mask those positions together and sample one replacement per position from the masked forward, at proposal temperature 1. **Gate:** form $\mathbf{x}^{\mathrm{all}}$ with every proposal inserted and run one unmasked forward.

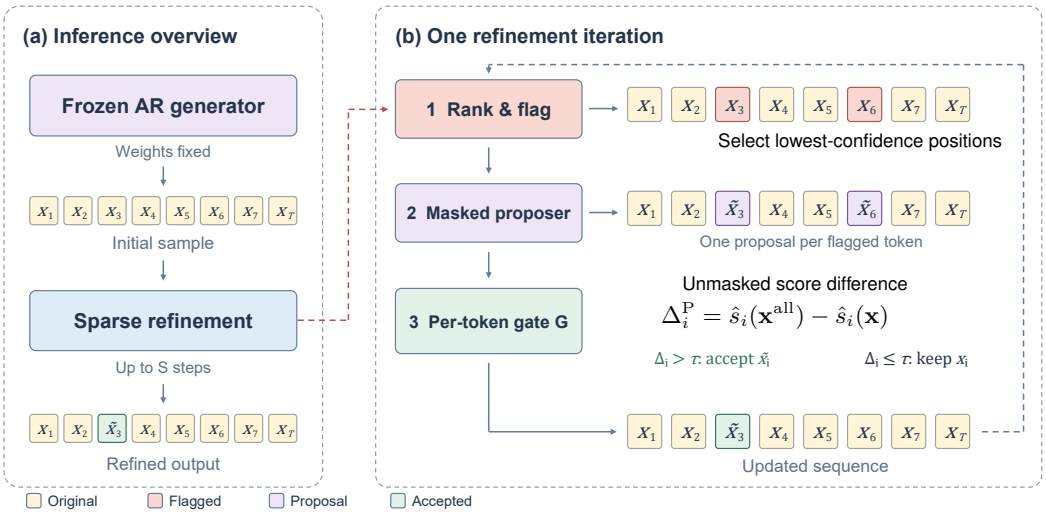

Figure 2: **The token-level gate.** Flagged images from Figure 1 enter the gate. Each iteration ranks positions, samples jointly masked proposals, and gates replacements using unmasked scores on the current and all-proposals sequences. Accepted edits are committed simultaneously; rejected positions retain their current tokens. The score comparison does not guarantee image-quality improvement.

For each selected position, compute

$$\Delta_i^{\mathrm{P}} = \hat{s}_i(\mathbf{x}^{\mathrm{all}}) - \hat{s}_i(\mathbf{x}), \qquad x_i^{\mathrm{next}} = \begin{cases} \tilde{x}_i, & \tilde{x}_i \neq x_i \text{ and } \Delta_i^{\mathrm{P}} > \tau, \\ x_i, & \text{otherwise.} \end{cases} \tag{2}$$

All decisions use the same two scored sequences and are committed in parallel. We use $S = 8$ iterations with a cosine mask-ratio schedule from 0.5 to 0.05, giving 24 refiner forwards per edited image.

The committed sequence generally differs from both scored sequences. Consequently, accepting a proposal does not imply improvement of a masked conditional, a joint density, PLL, or image quality. Sequential gating instead masks one queried position and compares alternatives under a fixed context; its local score property and conditional theoretical analysis appear in Appendices A.2 and A.3. It is an analyzable counterpart, not a guarantee inherited by the parallel gate used here, and Table 1 compares the two directly: the parallel gate has lower FID on all six degraded conditions (45.15 versus 64.10 on severe FFHQ), while the sequential gate stays closer to no repair on tuned inputs (7.83 versus 8.17 on FFHQ). The analysis therefore motivates the acceptance criterion; it does not certify the deployed variant.

3.3 Image-level routing and exact bypass

Let $g(\mathbf{x})$ be a scalar statistic of the initial sample, with sign $a \in \{-1, 1\}$ chosen so that larger ag indicates the severe side of the fitting distribution. The complete AGR policy is

$$\mathrm{AGR}(\mathbf{x}) = \begin{cases} \mathrm{Gate}(\mathbf{x}; \tau = 1), & ag(\mathbf{x}) \geq b, \\ \mathbf{x}, & ag(\mathbf{x}) < b. \end{cases} \tag{3}$$

By construction, a bypassed sample retains exactly its original tokens and therefore the same image under deterministic decoding. Flagged samples have no corresponding preservation or quality guarantee. Algorithm 1 states the complete procedure; the router reads its statistic from the same unmasked forward that the gate's first iteration uses for ranking, so a closed router costs one refiner forward.

The rule is fitted once per domain. We use $2{,}000$ tuned and $2{,}000$ severe validation samples, independently generated from the test split. Candidate statistics are mean current-token log-score $T^{-1}\sum_i \hat{s}_i(\mathbf{x})$, mean confidence $T^{-1}\sum_i c_i$, the fraction of first-step proposals with positive score difference, and the fraction of differing proposals accepted at $\tau = 0$. We select the statistic with highest validation tuned/severe AUROC and its cutoff by maximizing Youden's J (true-positive rate

Algorithm 1 Acceptance-Gated Refinement (router + parallel gate)

Require: token sequence $\mathbf{x} \in \mathcal{V}^T$ from the frozen generator; frozen refiner f_θ; router statistic g, sign a, cutoff b; gate steps S; mask-ratio schedule $\{\rho_s\}_{s=1}^{S}$; threshold τ
Ensure: output sequence $\mathbf{x}$

$\boldsymbol{\ell} \leftarrow f_\theta(\mathbf{x})$ ▷ Unmasked forward; gives $\hat{s}_i(\mathbf{x})$, c_i, and $g(\mathbf{x})$ (mean $\hat{s}_i$ or mean c_i)
if $a\, g(\mathbf{x}) < b$ **then**
  return $\mathbf{x}$ ▷ Router closed: exact bypass
end if
for $s = 1, \ldots, S$ **do**
  if $s > 1$ **then**
    $\boldsymbol{\ell} \leftarrow f_\theta(\mathbf{x})$ ▷ Re-rank on the current sequence
  end if
  $c_i \leftarrow \max_v \mathrm{softmax}(\boldsymbol{\ell}_i)_v$ for $i = 1, \ldots, T$
  $\mathcal{P}_s \leftarrow$ indices of the $\max(1, \lceil \rho_s T \rceil)$ smallest c_i ▷ Rank
  $\mathbf{x}^{\mathrm{mask}} \leftarrow \mathbf{x}$ with positions $\mathcal{P}_s$ set to `[MASK]`; $\mathbf{z} \leftarrow f_\theta(\mathbf{x}^{\mathrm{mask}})$
  $\tilde{x}_i \sim \mathrm{Categorical}(\mathrm{softmax}(\mathbf{z}_i))$ for $i \in \mathcal{P}_s$ ▷ Propose jointly, temperature 1
  $\mathbf{x}^{\mathrm{all}} \leftarrow \mathbf{x}$ with $\tilde{x}_i$ inserted for all $i \in \mathcal{P}_s$; $\boldsymbol{\ell}' \leftarrow f_\theta(\mathbf{x}^{\mathrm{all}})$ ▷ Score proposals
  $\Delta_i^{\mathrm{P}} \leftarrow \mathrm{logsoftmax}(\boldsymbol{\ell}'_i)[\tilde{x}_i] - \mathrm{logsoftmax}(\boldsymbol{\ell}_i)[x_i]$ for $i \in \mathcal{P}_s$ ▷ Gate score, Eq. (2)
  $\mathcal{A}_s \leftarrow \{\, i \in \mathcal{P}_s : \tilde{x}_i \neq x_i \text{ and } \Delta_i^{\mathrm{P}} > \tau \,\}$ ▷ Accepted positions
  $x_i \leftarrow \tilde{x}_i$ for all $i \in \mathcal{A}_s$ ▷ Commit simultaneously; other positions keep their tokens
end for
return $\mathbf{x}$

plus true-negative rate minus one). The statistic and cutoff are then frozen. Mild inputs and the additional intervention settings are excluded from router fitting. This is per-domain calibration, not transfer of a universal cutoff, but it is cheap: refitting the cutoff from only 25 tuned and 25 severe validation images opens on at most 1% of tuned and at least 96% of severe test images in every domain, and 100 per class is indistinguishable from the full 2,000 (Appendix C, Table 7).

Faces select mean log-score; flagging occurs at scores at most -7.470 on FFHQ and -7.198 on CelebA-HQ. ImageNet selects mean confidence and flags values at most 0.646 (full precision in Appendix C). Both selected statistics come from the initial unmasked pass, so the router adds no separate model. The evaluation pipeline collects all four candidate statistics using a three-forward first-step probe; computing only the selected statistic requires one forward. We report gate cost separately and do not infer an end-to-end speed advantage from these counts.

3.4 Relation to other operating policies

We compare AGR with three other operating policies. *Always-on gating* applies the gate to every input, either at one fixed τ or at a per-condition threshold selected from $\{0, 1, 2, 3, 4\}$ by validation FID; for the latter we distinguish forced refinement from selection with bypass allowed. *AGR without bypass* uses the same router but edits unflagged images at $\tau = 4$ instead of returning them; comparing it with Eq. (3) isolates the effect of the closed branch. None of these comparisons selects a threshold from the reported test FID. In summary, AGR differs from every comparison policy in one respect: its default action is to do nothing, so its cost on healthy inputs is set by the router's false-positive rate rather than by the gate's gentlest threshold.

4 Experiments

4.1 Setup and evaluation protocol

Main experiments use FFHQ, CelebA-HQ, and ImageNet at 256^2, with a LlamaGen VQ-16 tokenizer (16,384 entries, 256 tokens per image). The face generators are domain-trained GPT-B checkpoints; ImageNet uses the released GPT-B. Separate face refiners have 42.1M parameters and the class-conditional ImageNet refiner has 155.4M total parameters. Backbone coverage additionally includes GPT-L, GPT-XL at 384^2, RandAR-L, and a VQGAN transformer. The ImageNet B/L/RandAR experiments share a refiner; XL uses a sequence-length-specific refiner. Training details are in Appendix K.

Table 1: **Correction quality of correction procedures and operating policies on nine test conditions.** FID on 10k test samples per cell, lower is better. Mild and severe face inputs use temperatures 1.2 and 1.5; ImageNet tuned sampling uses cfg2 and its mild and severe tests use cfg1. "Label" marks procedures whose threshold is chosen with knowledge of the test condition. Bold marks the best entry in each column among procedures that modify samples; the shaded row is our full policy. "AGR without bypass" edits flagged images at $\tau = 1$ and the rest at $\tau = 4$. VSR is the class-conditioned control; random gating matches the calibrated gate's aggregate acceptance rate in expectation.

		FFHQ			CelebA-HQ			ImageNet		
Method	Label	Tuned	Mild	Severe	Tuned	Mild	Severe	Tuned	Mild	Severe
Reference										
No refinement	–	7.39	24.89	102.71	8.07	16.00	81.26	8.18	52.99	96.25
Selection and refinement baselines										
Best-of-4 (PLL select)	no	15.52	18.31	86.93	14.30	12.28	62.63	9.47	46.03	93.09
Ungated parallel (Chang et al., 2022)	no	26.94	33.11	57.71	19.90	21.60	41.27	33.23	49.85	**56.39**
VSR (Wang et al., 2025)	no	7.57	24.12	98.94	8.33	15.33	76.80	8.20	52.95	96.16
Random gate (rate-matched)	yes	7.92	29.23	79.18	8.71	23.56	62.18	8.21	47.18	69.58
Always-on gating, applied to every input										
Sequential gate	no	7.83	21.79	64.10	9.03	13.04	47.15	8.28	46.00	64.21
Parallel gate, fixed $\tau = 2$	no	14.37	18.87	55.00	14.58	11.95	37.32	8.95	44.08	69.87
Parallel gate, calibrated τ	yes	8.17	18.50	**45.15**	9.45	11.95	**27.67**	8.29	**41.82**	58.03
Gate with image-level routing (ours)										
AGR without bypass	no	8.28	**17.94**	47.37	9.55	**11.55**	30.17	8.28	44.21	58.07
AGR (router + gate + bypass)	no	**7.45**	18.55	47.36	**8.15**	12.31	30.17	**8.17**	44.92	58.09

We report clean-FID (Parmar et al., 2022) on 10,000 generated test images against full face reference sets or the 50k-image ImageNet validation reference. Router fitting uses the tuned/severe validation samples above. Threshold calibration separately uses 2k validation samples per condition; test samples are not used to fit either policy. ImageNet tuned sampling uses cfg2, while mild/severe tests use cfg1 at temperatures $1.2/1.5$: these are combined guidance/temperature shifts, not a fixed-guidance temperature sweep. Face mild/severe conditions use temperatures $1.2/1.5$. All comparisons within a condition start from the same input token bank. Mixtures contain 5k samples from each component.

Opening fractions are the proportion of images routed to repair, not token-level acceptance rates. Bootstrap standard deviations use 200 generated-side resamples with a fixed reference, and repair-seed repeats cover five gate conditions (Appendix D).

4.2 Main results: correction procedures and operating policies

Table 1 compares every correction procedure on nine test conditions: tuned, mild, and severe inputs in each of three domains. Rows are grouped by what a procedure assumes. The selection and refinement baselines and the always-on gates use no condition information. Per-condition calibration selects τ on validation samples of the same condition, so it knows which condition a test image comes from; it is a reference for the gate's attainable repair rather than a deployable policy. The random control is matched to that calibrated gate. The two routed policies in the last group see only the image. Two rows are references rather than competitors: the tuned column of "no refinement" is what regenerating at known-good settings would give, so no procedure is expected to beat it, and "ungated parallel" is the gate's $\tau \to -\infty$ limit, included to show what the acceptance criterion adds.

Three observations organize the rest of this section. First, on every tuned column AGR is the best procedure that modifies samples and is within 0.08 FID of no refinement, whereas always-on gating costs between 0.10 and 6.98 FID there and ungated refinement costs up to 25.05. Second, on severe columns AGR retains most of the repair that calibration reaches with condition labels (47.36 versus 45.15 on FFHQ, 30.17 versus 27.67 on CelebA-HQ, and 58.09 versus 58.03 on ImageNet) and matches AGR without bypass within 0.02. Third, no procedure dominates: best-of-four remains competitive on mild FFHQ (18.31), ungated parallel refinement is the best severe ImageNet entry (56.39) while raising tuned ImageNet FID from 8.18 to 33.23, and VSR is approximately unchanged

Table 2: **Decomposition of AGR on individual regimes.** FID on 10k images. "Always-on gate" applies the gate to every image at the per-condition calibrated threshold, which uses the condition label; the routed policies share the router of Section 3.3 and differ only in the closed branch, which AGR without bypass edits at $\tau = 4$ and AGR returns exactly. "Open" is the fraction of images sent to the gate at $\tau = 1$. Bold marks the best entry per column among procedures that modify samples; the shaded row is our full policy.

	FFHQ			CelebA-HQ			ImageNet		
Policy	Tuned	Mild	Severe	Tuned	Mild	Severe	Tuned	Mild	Severe
No repair	7.39	24.89	102.71	8.07	16.00	81.26	8.18	52.99	96.25
Always-on gate, calibrated τ	8.17	18.50	**45.15**	9.45	11.95	**27.67**	8.29	**41.82**	**58.03**
AGR without bypass	8.28	**17.94**	47.37	9.55	**11.55**	30.17	8.28	44.21	58.07
AGR	**7.45**	18.55	47.36	**8.15**	12.31	30.17	**8.17**	44.92	58.09
Open (%)	1.5	41.4	99.5	1.3	32.2	99.6	0.4	51.6	99.6



Figure 3: **Paired ImageNet correction examples.** Each pair shows the same GPT-B sample before (left) and after (right) the gate. Rows show cfg2 sampling and cfg1 sampling at temperatures 1.0, 1.2, and 1.5. These fixed-setting gate examples illustrate the scope of correction; outputs of the routed policy itself are shown in Figure 9 (Appendix H).

everywhere. Score-gated procedures are the only rows that improve degraded inputs without large tuned-input damage, and AGR alone does so without condition information.

4.3 Ablation: the closed branch

Table 2 decomposes the policy. Reading down its rows: the always-on gate repairs degraded inputs but costs 0.11–1.38 FID on tuned ones and needs the condition label to pick its threshold; adding the router removes the label; and the two routed rows differ only in the closed branch, which AGR without bypass edits at $\tau = 4$ and AGR returns unchanged. Figure 3 shows paired gate outputs across ImageNet conditions.

On tuned inputs AGR bypasses 98.5%, 98.7%, and 99.6% of FFHQ, CelebA-HQ, and ImageNet inputs (Table 2). Their observed FID changes are $+0.06$, $+0.08$, and -0.01. This improves preservation over AGR without bypass, whose tuned face FIDs are 8.28 and 9.55. In comparison, always-on gating at $\tau = 2$ yields 14.37 and 14.58. Bypass prevents changes to the unflagged majority; the flagged 0.4–1.5% can still be harmed, which is why we report the aggregate change rather than claim zero cost. The closed branch is also cheap: measured end to end on the same boundary as our other timings (Table 11, Appendix G), AGR adds 0.1% at batch 1 and 1.7% at batch 25 to generation time on tuned input, against 2.3% and 27.6% for always-on gating; on a 50/50 mixture it adds 1.3% and 15.1%.

On degraded inputs the picture reverses. Over 99% of severe inputs open the gate, reducing FID from 102.71 to 47.36 on FFHQ, 81.26 to 30.17 on CelebA-HQ, and 96.25 to 58.09 on ImageNet. The fixed open-branch threshold $\tau = 1$ leaves some repair unused: condition-calibrated $\tau = 0$ reaches 45.15 and 27.67 on severe faces. Mild inputs, unseen during router fitting, also improve at opening

Table 3: **Ablation on mixed inputs.** Each row contains 5k tuned and 5k severe (A) or mild (B) inputs. "Regime" chooses $\tau \in \{1, 4\}$ per condition using validation FID; it has regime labels and is not an oracle bound. "w/o bypass" is AGR without bypass. Bold marks the best policy that modifies samples per row; small differences are descriptive.

Mixture	No repair	$\tau = 1$	$\tau = 2$	$\tau = 4$	Regime	w/o bypass	AGR
FFHQ							
A: tuned + severe	37.56	23.28	22.33	30.79	18.95	19.02	**18.72**
B: tuned + mild	12.06	18.68	13.90	10.83	10.83	10.71	**10.56**
CelebA-HQ							
A: tuned + severe	29.65	15.38	15.85	23.39	13.63	13.67	**13.57**
B: tuned + mild	9.44	14.84	11.71	**8.84**	10.91	9.23	8.92
ImageNet							
A: tuned + severe	34.58	24.55	25.93	32.56	21.69	21.73	**21.68**
B: tuned + mild	19.86	19.56	17.83	19.21	**16.71**	17.18	17.29

Table 4: **Router and gate under held-out tests.** (a) Frozen routers on settings excluded from fitting: routers are fitted only on tuned/severe temperature conditions, and forced gating uses the displayed threshold. Exact bypass avoids further changes to truncated samples but leaves their degradation untreated. (b) Random editing at the calibrated target acceptance rate (0.055, 0.125, and 0.453 for the three rows): the control matches aggregate edit rates in expectation, not per-image counts, and all methods start from the same input bank. Bold marks the better policy per row; shaded columns are ours.

(a) Settings excluded from router fitting

Intervention	No repair	AGR	Open (%)	Forced gate
FFHQ, 3-bit weights[†]	285.69			**176.90** (τ=0)
FFHQ, 4-bit weights	33.02	31.68	10.9	**30.71** (τ=3)
FFHQ, top-$k = 16$	73.26	**73.26**	0.0	74.11 (τ=4)
FFHQ, top-$p = 0.80$	27.48	**27.48**	0.0	29.28 (τ=4)
ImageNet, no CFG, $t = 1$	26.07	26.11	1.8	**25.92** (τ=4)

(b) Random editing at the calibrated rate

Condition	τ	No repair	Gate	Random
FFHQ mild	3	24.89	**18.50**	29.23
CelebA-HQ mild	2	16.00	**11.95**	23.56
FFHQ severe	0	102.71	**45.15**	79.18

[†] For 3-bit, only the verified test-set forced-gate result is reported; no split-matched routed result is reported here.

fractions of 32–52%, with bypass trading some mild-regime quality for tuned-input preservation (threshold-sweep precision and recall in Appendix H).

4.4 Ablation: heterogeneous mixtures without corruption labels

Table 3 evaluates the frozen rules on unlabeled mixtures, the operating condition the router is meant for. On every tuned/severe mixture AGR is the best policy tested: 18.72, 13.57, and 21.68, against 37.56, 29.65, and 34.58 without repair, and marginally below the regime-aware reference (18.95, 13.63, 21.69), which uses regime labels but only selects between two thresholds. On tuned/mild mixtures it is best on FFHQ (10.56), within 0.1 of fixed $\tau = 4$ on CelebA-HQ (8.92 versus 8.84), and within 0.6 of the regime-aware reference on ImageNet (17.29 versus 16.71).

No fixed threshold is competitive on both mixtures of any domain. $\tau = 4$ is best on the CelebA-HQ tuned/mild mixture but the worst edited policy on every tuned/severe mixture, and $\tau = 1$ improves every tuned/severe mixture but worsens FFHQ tuned/mild from 12.06 to 18.68. AGR is the only policy that is never worse than no repair on any mixture and best or near-best on all six. The margins among the leading policies are within the bootstrap SD of about 0.3 (Appendix D), and we do not claim them. What the closed branch buys is not mixture FID but a guarantee: on the 98.5% or more of tuned images the router leaves alone, the output tokens are the input tokens, which no always-on or two-threshold policy can offer.

4.5 Unseen interventions: the router opens where the gate helps

Why does a router fitted only on temperature transfer to damage it never saw? Figure 8 (Appendix H) gives the reason. Across thirteen settings in three domains, whether the gate helps is decided by whether the input's refiner score falls below the domain's tuned samples, and up to the severe temperature condition the size of the gain grows with that deficit. Quantization at 4 and 3 bits, which

the router never saw, falls on the same side of the FFHQ curve as the temperature conditions it was fitted on. Top-k and top-p truncation sit on the other side of zero: their samples score *higher* than tuned ones, the refiner sees nothing locally implausible to fix, and forcing the gate makes them worse. The router thresholds exactly this deficit, so it is not a temperature detector; it opens where the gate has something to fix and stays closed where it does not.

Table 4a confirms this per image on the held-out settings. The router applies correction to 10.9% of 4-bit inputs and improves FID from 33.02 to 31.68, against 30.71 for calibration that uses quantization samples to pick its threshold. It opens on none of the truncated samples, where even the strictest forced threshold worsens FID, and on 1.8% of guidance-free ImageNet inputs, where calibration at $\tau = 4$ gives 25.92 against a 26.07 baseline while permissive refinement at $\tau = 0$ gives 49.07. In every case the per-image decision agrees with the verdict of an independent validation sweep for that setting. Abstention prevents additional damage; it does not turn poor samples into good ones, and improvement on a flagged population does not show that each flagged image benefits.

4.6 Why score-based token selection matters

The random control keeps the gate's ranking and masked proposals but accepts each differing proposal with probability $\min(1, R\, n_{\text{prop}}/n_{\text{diff}})$, where R is the calibrated gate's measured aggregate acceptance rate; trajectories diverge after the first step, so proposals are not paired across runs.

At these matched target rates, random editing worsens mild FFHQ and CelebA-HQ relative to no repair, while score gating improves both (Table 4b); on severe FFHQ, score gating is better by 34.03 FID. With the edit budget held fixed, the acceptance criterion is what separates improvement from damage. The control matches the aggregate rate rather than per-image counts, so we do not attribute every unit of the difference to it.

The gate also transfers across backbones. Validation-calibrated gating repairs severe GPT-L from 78.80 to 52.27, GPT-XL at 384^2 from 140.19 to 79.95, and RandAR-L from 35.98 to 24.70 (Table 5, Appendix B). These extend the gate evidence, not the set of backbones on which the router is evaluated. A single fixed threshold transfers unevenly across refiners. The class-conditioned VSR control is approximately unchanged on ImageNet (8.20, 52.95, 96.16 for tuned/mild/severe) and does not reproduce the gate's repair gains.

5 Discussion and Conclusion

Selective correction requires an image-level intervention decision and a token-level commitment rule. Across three domains, AGR bypasses most tuned samples at negligible FID and time cost, retains substantial repair on degraded inputs, and opens on unseen quantized inputs while staying closed on truncated ones, because both the router and the gate respond to the same score deficit. Bypass is what makes the policy safe to leave on: conservative refinement still edits normal samples, whereas a closed router preserves them exactly, at the cost of one refiner forward.

Several limitations bound these claims. The router needs tuned and severe validation inputs, although as few as 25 per class suffice, and a separate fit per domain, and evaluation covers chosen generation shifts and $50/50$ mixtures, not arbitrary deployment failures or prevalence. False positives can harm normal samples, false negatives leave damage untreated, and the $\tau = 1$ repair branch need not be optimal. Unmasked refiner scores are not calibrated probabilities of repair success, and small FID differences do not establish equivalence or per-image preservation. Refiner training and inference add cost, and nothing here shows an efficiency advantage over regenerating when regeneration is possible.

Regarding broader impacts, correction can alter appearance or amplify biases of the generator and refiner, and neither the gate nor the router evaluates identity, content safety, or misuse; no safety or energy benefit follows from the reported FID gains.

Reproducibility Statement

Section 3 specifies the scoring functions, routing cutoffs, branches, and gate settings. Section 4.1 distinguishes router fitting, threshold selection, and test evaluation. Appendices B–D detail policy

comparisons, mixture construction, and reported uncertainty; Appendices E and K give calibration records and refiner configurations. The sequential analysis states its assumptions explicitly in Appendix A.3. Code, refiner checkpoints, routing cutoffs, and per-run result records will be released.

AI USE STATEMENT

Generative AI tools assisted with manuscript organization and editing, experimental-design feedback, interpretation of results, and figure drafting and conversion to editable artwork. These uses do not constitute independent reproduction of the reported experiments.

REFERENCES

Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. *arXiv preprint arXiv:1810.06758*, 2018.

Julian Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.

Victor Besnier, David Hurych, Andrei Bursuc, and Eduardo Valle. BIGFix: Bidirectional image generation with token fixing. *arXiv preprint arXiv:2510.12231*, 2025.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. In *International Conference on Machine Learning (ICML)*, 2023.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Cheng Cheng et al. TensorAR: Refinement is all you need in autoregressive image generation. *arXiv preprint arXiv:2505.16324*, 2025.

Rewon Child. Very deep VAEs generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations (ICLR)*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Jerry Tworek, Jacob Hilton, Rei Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. In *arXiv preprint arXiv:2110.14168*, 2021.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Doohyuk Jang, Sihwan Park, June Yong Yang, Yeonsung Jung, Jihun Yun, Souvik Kundu, Sung-Yub Kim, and Eunho Yang. Lantern: Accelerating visual autoregressive models with relaxed speculative decoding. *International Conference on Learning Representations (ICLR)*, 2025.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. *International Conference on Machine Learning (ICML)*, 2023.

José Lezama, Huiwen Chang, Lu Lang, and Irfan Essa. Improved masked image generation with token-critic. In *European Conference on Computer Vision (ECCV)*, 2022.

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

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T. Freeman, and Yu-Xiong Wang. RandAR: Decoder-only autoregressive visual generation in random orders. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

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 (NeurIPS)*, 2024.

Ryan Turner, Jane Hung, Eric Frank, Yunus Saatci, and Jason Yosinski. Metropolis-Hastings generative adversarial networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jiamian Wang et al. Visual self-refinement for autoregressive models. In *Findings of the Association for Computational Linguistics: EMNLP*, 2025.

A WHY NAIVE REFINEMENT FAILS AND WHAT THE GATE GUARANTEES

This appendix collects the motivation and analysis behind the acceptance gate. Section A.1 gives the three diagnostics that argue against ungated or continuous refinement, Section A.2 defines the sequential gate and its local score guarantee, and Section A.3 states the idealized results that hold when the refiner's conditionals are mutually compatible. Throughout, "the sequential gate" masks one queried position and compares alternatives under a fixed context; the parallel gate of the main text inherits none of these guarantees.

A.1 WHY NAIVE POST-HOC REFINEMENT FAILS

A refiner's preference is not itself evidence of image-quality improvement. Three diagnostics motivate discrete proposals filtered by an explicit acceptance test (Figure 4).

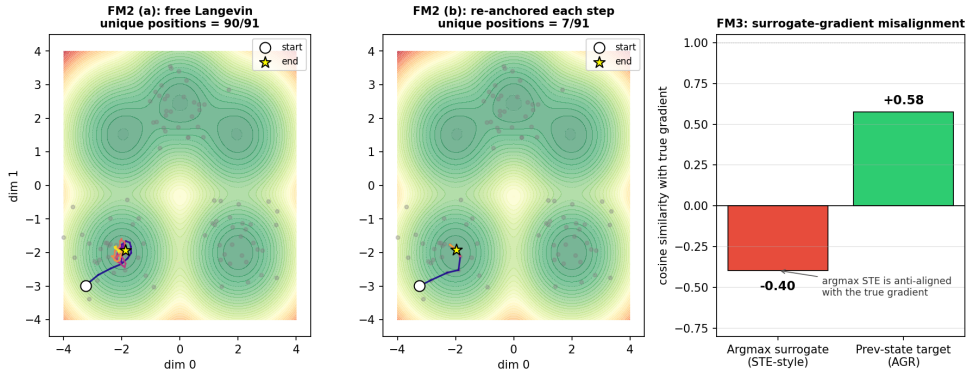

Figure 4: **Failure diagnostics for naive post-hoc refinement.** (a) Greedy discrete replacement can reinforce its own earlier mistakes, so confidence rises while sequence quality falls. (b) Continuous refinement loses most of its apparent exploration once each step is projected back to a discrete codebook. (c) The surrogate gradient available to an embedding-space corrector is poorly aligned with the oracle correction direction.

FM1: ungated edits reinforce errors. When every proposal is committed, it becomes context for later edits. On the structured toy benchmark of Appendix J, greedy replacement changes 86.7% of tokens and reduces rule accuracy from 0.946 to 0.619, with no fully valid output sequence. On modern generators, ungated parallel refinement damages tuned samples: FFHQ FID rises from 7.39 to 26.94 (Table 1).

FM2: discretization collapses continuous trajectories. We run inertial Langevin dynamics for 90 steps on a two-dimensional five-mode Gaussian-mixture energy (shared width $\sigma = 0.8$) over a codebook-like set of 90 irregular anchors drawn from three clusters, and compare free dynamics with projection to the nearest anchor after every step. Free dynamics visit 89 distinct positions; every-step projection visits 8, spending most of the budget cycling among a handful of neighbors. The order of magnitude, not the exact count, is the point.

FM3: the refiner's gradient is not a correction signal. We train a small masked-denoising refiner ($|\mathcal{V}| = 32$, $T = 16$, $d = 32$, two layers) on synthetic sequences, corrupt 40% of a clean sequence, and at the corrupted embedding compare three gradients of the masked-denoising loss at six probe positions: toward the refiner's current argmax prediction (the direction available at inference), toward the current corrupted token, and toward the clean token (the oracle direction, used only as a reference). The inference-time gradient has cosine similarity 0.08 with the oracle direction, versus 0.66 for the gradient between the two known token states. The refiner is trained to predict masked tokens from context, not to define a smooth correction energy in embedding space. Both numbers depend on the refiner and probe; the ordering was stable across our exploratory runs, and we use it as a structural argument rather than a precise quantity.

These diagnostics motivate operating on discrete tokens with stochastic proposals and rejecting edits that fail a local score test. They do not show that a refiner-derived test detects every harmful edit; the

proposer and gate share one learned refiner, and Section 4.5 tests the remaining risk that accepted edits raise the score while worsening the image.

A.2 THE SEQUENTIAL GATE AND ITS LOCAL GUARANTEE

For a position i, let $\mathbf{x}_{\mathrm{mask}_i}$ be the sequence with x_i replaced by the mask token and let $x_{\setminus i} := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_T)$ be the context. The refiner defines a single-site conditional and local log-score

$$q_i(v \mid x_{\setminus i}) := \mathrm{softmax}\big(f_\theta(\mathbf{x}_{\mathrm{mask}_i})_{[i]}\big)_v, \qquad s_i(v; x_{\setminus i}) := \log q_i(v \mid x_{\setminus i}). \tag{4}$$

The sequential gate runs S steps; in each step it selects the low-confidence positions, and for each selected position in turn draws a proposal $\tilde{x}_i \sim \mathrm{softmax}(\cdot/\beta)$ from the masked logits and computes the local conditional gain

$$\Delta_i = s_i(\tilde{x}_i; x_{\setminus i}) - s_i(x_i; x_{\setminus i}), \qquad \text{accepting the edit iff} \qquad \Delta_i > \tau. \tag{5}$$

Proposal and acceptance score come from the same masked forward, so the gate adds no forward over ungated resampling; accepted edits enter the context of later positions within the step. Each step costs one unmasked pass plus one masked pass per selected position.

Proposition 1 (Local conditional improvement)**.** *If the sequential gate accepts a proposal replacing x_i by $\tilde{x}_i$ while $x_{\setminus i}$ is held fixed, then $s_i(\tilde{x}_i; x_{\setminus i}) > s_i(x_i; x_{\setminus i}) + \tau$.*

Proof. Immediate from Eq. (5). □

This is a property of the rule, not an image-quality guarantee: it concerns one token under a fixed context and implies neither better FID nor an increase of the refiner's pseudo-log-likelihood

$$\mathcal{S}(\mathbf{x}) = \sum_{i=1}^{T} s_i(x_i; x_{\setminus i}), \tag{6}$$

since editing x_i changes every other term's context. Appendix I.2 nonetheless observes a PLL increase on all $50{,}000$ legacy FFHQ sequences, and Section 4.5 shows that even a PLL increase need not mean better images.

A.3 FORMAL ANALYSIS UNDER COMPATIBLE CONDITIONALS

The local gain acquires a joint-density meaning when the refiner's conditionals are those of one normalized distribution, which masked denoising does not enforce (Besag, 1975; Salazar et al., 2020).

Assumption 1 (Compatible conditionals)**.** *There is a strictly positive joint distribution q on $\mathcal{V}^T$ with $q(x_i \mid x_{\setminus i}) = q_i(x_i \mid x_{\setminus i})$ for every i and $\mathbf{x}$.*

Lemma 1 (Conditional gain equals joint gain)**.** *Under Assumption 1, if $\mathbf{x}'$ differs from $\mathbf{x}$ only at position i, then $\log q(\mathbf{x}') - \log q(\mathbf{x}) = \Delta_i$.*

Proof. $\log q(\mathbf{x}) = \log q(x_i \mid x_{\setminus i}) + \log q(x_{\setminus i})$ and likewise for $\mathbf{x}'$; the shared marginal $\log q(x_{\setminus i})$ cancels and the assumption replaces each conditional by q_i. □

Proposition 2 (Monotone joint ascent and finite edits)**.** *Under Assumption 1, every accepted edit increases $\log q(\mathbf{x})$ by more than τ. Consequently, for $\tau > 0$, an idealized process that keeps proposing single-site edits accepts at most $N < \big(\max_{\mathbf{x}} \log q(\mathbf{x}) - \log q(\mathbf{x}_0)\big)/\tau$ edits.*

Proof. The first claim is Lemma 1 with $\Delta_i > \tau$. For the second, $\mathcal{V}^T$ is finite and q is positive, so $M := \max_{\mathbf{x}} \log q(\mathbf{x}) < \infty$; after N accepted edits $\log q(\mathbf{x}_N) > \log q(\mathbf{x}_0) + N\tau$, and $\log q(\mathbf{x}_N) \le M$ bounds N. □

Assumption 2 (Fair repeated proposals)**.** *Every single-site edit with positive proposal probability is proposed infinitely often unless the process terminates first; a full-support softmax proposal with $\beta > 0$ visiting every position suffices.*

Proposition 3 (τ-local optimality)**.** *Under Assumptions 1 and 2 with $\tau > 0$, the idealized process almost surely reaches a state $\mathbf{x}^\star$ such that every single-site edit $\mathbf{x}^\star \to \mathbf{y}$ in the proposal support satisfies $\log q(\mathbf{y}) - \log q(\mathbf{x}^\star) \le \tau$.*

Proof. By Proposition 2 the state is eventually fixed at some $\mathbf{x}^\star$. If an edit with joint gain above τ existed in the support, Lemma 1 gives $\Delta_i > \tau$, fairness proposes it after the stopping time, and it would be accepted, a contradiction. □

These results concern $\log q$, not the pseudo-log-likelihood of Eq. (6), whose overlapping contexts make monotonicity an empirical question. The gate also shares only the proposal–acceptance structure of Metropolis–Hastings (Metropolis et al., 1953; Hastings, 1970): it omits the proposal-probability ratio and rejects every non-improving proposal, so it is a thresholded score-ascent procedure with neither detailed balance nor exact sampling. The empirical repairability claims of the main text do not rely on any statement in this subsection.

B Complete Gate and Operating-Policy Comparisons

Table 1 retains the full comparison among correction procedures. AGR and AGR without bypass use identical routing criteria but different closed branches. Calibrated gate rows report forced refinement at the validation-selected threshold; they do not replace tuned values with bypass. The random-control row is matched to the calibrated acceptance rate on mild FFHQ and CelebA-HQ. VSR uses the class-conditioned variant on ImageNet. The comparison therefore separates the choice of gate from the image-level operating policy.

Backbone coverage. Table 5 evaluates the gate across additional backbones. Calibration is performed separately per condition, and a fixed threshold is not equally effective across tokenizer/refiner pairs. These results do not evaluate bypass routing on every listed backbone.

Table 5: **Backbone coverage and operating-point sensitivity.** FID on 10k samples. "Cal." selects τ on validation data; "Fixed" uses $\tau = 2$. Bold marks the better gate setting per row and regime.

	Tuned			Severe		
Backbone	No repair	Cal.	Fixed	No repair	Cal.	Fixed
LlamaGen GPT-B (Sun et al., 2024)	8.18	**8.29**	8.95	96.25	**58.03**	69.87
LlamaGen GPT-L (Sun et al., 2024)	8.01	**8.05**	8.32	78.80	**52.27**	60.27
LlamaGen GPT-XL, 384^2 (Sun et al., 2024)	6.70	**6.71**	7.24	140.19	**79.95**	82.62
VQGAN transformer (Esser et al., 2021)	16.92	**16.82**	16.94	37.38	**25.88**	37.13
RandAR-L (Pang et al., 2025)	5.87	**5.90**	6.28	35.98	**24.70**	**24.70**

Evaluation-protocol anchor. An additional 50k-image ImageNet GPT-B cfg2 evaluation using legacy-TensorFlow Inception features and the ADM reference moments gives FID 5.237 without refinement and 5.385 with forced gating at $\tau = 4$. These values are separate from the 10k clean-FID protocol used in the main comparisons and do not demonstrate exact no-op behavior.

C Adaptive-Policy and Mixture Details

The selected statistics are computed on the original input before any committed edit. For a selected position in the first iteration, let Δ_i^{P} be the unmasked score difference in Eq. (2). The four candidates are mean current-token log-score, mean maximum probability, the mean indicator $\mathbb{1}[\Delta_i^{\mathrm{P}} > 0]$ over proposed positions, and the mean indicator $\mathbb{1}[\Delta_i^{\mathrm{P}} > 0, \tilde{x}_i \neq x_i]$. The first two require only the initial unmasked pass; the latter two require masked proposals and an all-proposals score pass.

The fitting procedure compares orientation-invariant validation AUROCs rounded to four decimals, with candidate order breaking ties. The sign is determined by the difference in validation medians between severe and tuned samples. It evaluates unique signed validation scores as candidate cutoffs and chooses the first maximizer of Youden's J. The exact cutoffs are -7.470152378082275 for FFHQ mean log-score, -7.197504997253418 for CelebA-HQ mean log-score, and 0.6464734673500061

for ImageNet mean confidence, with lower scores opening the gate. AUROCs use sorting ranks rather than tie-averaged ranks.

Table 6: Selected routing statistics and recorded discrimination on held-out test samples. Mild conditions are excluded from fitting. AUROC measures regime discrimination, not the probability that an edit helps.

Domain	Statistic	Tuned/severe AUROC	Tuned/mild AUROC
FFHQ	Mean log-score	0.9992	0.9401
CelebA-HQ	Mean log-score	0.9995	0.9293
ImageNet	Mean confidence	0.9999	0.9789

Mixture A concatenates tuned indices $0{:}5000$ with severe indices $0{:}5000$. Mixture B concatenates tuned indices $5000{:}10000$ with mild indices $0{:}5000$. Thus the tuned components are disjoint between the two mixtures. Routing operates per image without the component label. The regime-aware reference selects between $\tau = 1$ and $\tau = 4$ using each regime's own validation FID, including mild data unavailable to router fitting. It selects $(4, 4, 1)$ for FFHQ tuned/mild/severe and $(4, 1, 1)$ for CelebA-HQ and ImageNet. This restricted reference neither allows bypass nor directly minimizes mixture FID.

Calibration sample efficiency. Table 7 refits the cutoff by Youden's J on n randomly drawn validation images per class, $n \in \{25, 50, 100, 500, 2000\}$ with 200 draws per n ($n = 2000$ is the full split and therefore the paper's rule), and applies each refit cutoff to the full 10k test split. For FFHQ, whose Inception features are stored, the resulting AGR FID is also recomputed (40 draws per n). Twenty-five images per class already give deployment-grade behavior: in every domain the cutoff opens on at most 1% of tuned and at least 96% of severe test images, and on FFHQ the AGR FID is within 0.03 (tuned), 0.5 (severe), and 0.15 (mixture A) of the full calibration. At 100 per class the cutoff standard deviation is at most 0.08 nats (faces) or 0.013 (ImageNet confidence) and every FFHQ FID is within 0.1 of the full calibration. The only quantity that moves with n is the opening fraction on mild inputs, which sit near the cutoff and were never part of the fit; it swings by about ± 12 points at $n \leq 100$ and ± 6–8 at $n = 500$. Small-n cutoffs are biased toward the closed side, so an under-sampled router under-opens rather than over-opens.

Table 7: Router refit from n validation images per class. Mean $\pm$ sd over draws; cutoffs in $-\log p$ for faces and confidence for ImageNet; opened fractions on the 10k test split; AGR FID on FFHQ only, where stored features allow recomputation.

n	Cutoff	Open tuned (%)	Open mild (%)	Open severe (%)	FID tuned	FID severe	FID mixture A
FFHQ							
25	7.596 ± 0.099	0.7 ± 0.8	25.0 ± 12.5	96.0 ± 3.9	7.42 ± 0.03	47.89 ± 1.02	18.87 ± 0.31
50	7.538 ± 0.090	1.0 ± 1.0	32.0 ± 12.8	98.0 ± 1.8	7.44 ± 0.06	47.55 ± 0.18	18.78 ± 0.04
100	7.490 ± 0.081	1.4 ± 1.0	38.8 ± 12.0	99.0 ± 1.0	7.45 ± 0.05	47.46 ± 0.09	18.76 ± 0.03
500	7.487 ± 0.037	1.3 ± 0.4	39.0 ± 5.8	99.3 ± 0.4	7.44 ± 0.02	47.40 ± 0.05	18.73 ± 0.02
2000	7.470	1.5	41.4	99.5	7.45	47.36	18.71
CelebA-HQ							
25	7.345 ± 0.098	0.5 ± 0.5	18.4 ± 8.6	96.8 ± 2.7	–	–	–
50	7.310 ± 0.084	0.6 ± 0.4	21.2 ± 7.9	97.8 ± 1.8	–	–	–
100	7.258 ± 0.070	0.9 ± 0.5	26.1 ± 7.2	98.8 ± 1.1	–	–	–
500	7.208 ± 0.027	1.2 ± 0.2	31.1 ± 3.1	99.5 ± 0.2	–	–	–
2000	7.198	1.3	32.2	99.5	–	–	–
ImageNet							
25	-0.624 ± 0.015	0.1 ± 0.2	31.2 ± 13.1	96.5 ± 3.9	–	–	–
50	-0.633 ± 0.014	0.2 ± 0.2	38.6 ± 12.8	98.2 ± 1.6	–	–	–
100	-0.639 ± 0.013	0.3 ± 0.3	44.4 ± 12.2	98.9 ± 1.0	–	–	–
500	-0.649 ± 0.008	0.5 ± 0.3	54.2 ± 7.5	99.6 ± 0.2	–	–	–
2000	-0.646	0.4	51.6	99.6	–	–	–

Cost accounting. The first-step evaluation probe performs three forwards per image, and a complete $S = 8$ gate performs 24. These are component counts, not a measured total latency for the routed policy. An implementation that runs both components independently uses $3 + 24p$ forwards on average when the opening fraction is p; reusing the probe as the first gate iteration instead gives

$3 + 21p$, and computing only the selected statistic from the ranking forward, as in Algorithm 1, gives $1 + 23p$. Measured end-to-end timings for the deployed policy are in Table 11. These implementation choices require consistent batching and timing before making an end-to-end cost comparison.

D Reported Sampling and Repair Variability

The feature-based analysis recomputes FID from stored Inception features and resamples the generated side 200 times, treating reference moments as fixed. Table 8 reports point estimates and bootstrap standard deviations separately. These SDs describe variation under that resampling scheme, not uncertainty over refiner training or all generation seeds. They are not confidence intervals for paired method differences. The bootstrap mean is upward-shifted relative to the original point estimate, so we do not interpret an unadjusted percentile interval as an equivalence test or compare a difference to one method's SD to declare significance.

Table 8: Generated-side bootstrap variability for AGR on FFHQ. Reference statistics are fixed; $B = 200$ resamples per entry. The mixture rows use 5k images from each component before resampling.

Input	AGR FID	Bootstrap SD
Tuned	7.447	0.128
Mild	18.553	0.239
Severe	47.362	0.340
Tuned + severe	18.715	0.293
Tuned + mild	10.560	0.209

Separate repeated-repair-seed results in Table 9 keep the initial token bank fixed and vary refinement randomness. They therefore address a different source of variation. Small differences among policies are reported descriptively; exact token preservation is claimed only on the bypass branch of Eq. (3).

E Validation Calibration Records

The selected threshold uses a disjoint 2k validation split and is scored on 10k test samples. The ImageNet severe seed replicates use a later seed-0 rerun (57.88); the gate comparison tables report the original run (58.03). The refine/bypass verdict is separate from the selected refinement threshold. Verdicts with margins below 0.2 are descriptive, not equivalence tests. Where two runs of the same configuration exist (for example 74.83 here versus 74.84 in the timing table for speculative $\delta = 0.03$), the values are independent reruns on different hardware; Table 9 reports seed replicates for five cells.

Table 9: Seed replicates (10k samples each) for four main-table cells and one backbone-coverage cell. Seed 0 is the run in the replicate series; mean $\pm$ sample standard deviation over the three runs.

Cell	Policy	Seed 0	Seed 1	Seed 2	Mean $\pm$ sd
FFHQ $t = 1.5$	gate $\tau = 0$	45.15	44.77	45.22	45.05 ± 0.24
FFHQ $t = 1.2$	gate $\tau = 3$	18.50	18.80	18.58	18.63 ± 0.15
CelebA-HQ $t = 1.5$	gate $\tau = 0$	27.67	27.33	27.77	27.59 ± 0.23
ImageNet cfg1, $t = 1.5$	gate $\tau = 1$	57.88	57.65	57.70	57.75 ± 0.12
RandAR-L severe	gate $\tau = 2$	24.70	24.58	24.64	24.64 ± 0.06

Table 10: Validation-selected thresholds and test FID. "Earlier" is the original operating point; "Calibrated" is the best refinement threshold on validation and its test FID; "Val. margin" is that threshold's validation FID minus the bypass validation FID, so negative favors refinement and the verdict is its sign. Cell names: `t` temperature, `w` weight bits, `cfg` guidance, `l256` GPT-L, `xl384v2` GPT-XL at 384^2, `randarv2` RandAR-L, `vq_bal` the VQGAN transformer, `lf` relaxed speculative decoding at the given δ (Appendix G).

Condition	Base	Earlier τ/FID	Calibrated τ/FID	Test diff.	Val. margin	Verdict
ffhq_t10	7.39	$\tau 4 \to 8.17$	$\tau 4 \to 8.17$	+0.00	+0.72	bypass
ffhq_t12	24.89	$\tau 1 \to 21.62$	$\tau 3 \to 18.50$	−3.12	−5.81	refine
ffhq_t15	102.71	$\tau 0 \to 45.15$	$\tau 0 \to 45.15$	+0.00	−54.69	refine
ffhq_w4	33.02	$\tau 1 \to 34.35$	$\tau 3 \to 30.71$	−3.64	−2.84	refine
ffhq_topk16	73.26	$\tau 1 \to 78.79$	$\tau 4 \to 74.11$	−4.68	+0.87	bypass
ffhq_topp80	27.48	$\tau 1 \to 40.19$	$\tau 4 \to 29.28$	−10.91	+1.61	bypass
celeba_t10	8.07	$\tau 4 \to 9.45$	$\tau 4 \to 9.45$	+0.00	+1.26	bypass
celeba_t12	16.00	$\tau 1 \to 13.95$	$\tau 2 \to 11.95$	−2.00	−5.00	refine
celeba_t15	81.26	$\tau 0 \to 27.67$	$\tau 0 \to 27.67$	+0.00	−54.91	refine
in_cfg2	8.18	$\tau 4 \to 8.29$	$\tau 4 \to 8.29$	+0.00	+0.10	bypass
in_cfg1_t12	52.99	$\tau 1 \to 41.82$	$\tau 1 \to 41.82$	+0.00	−10.83	refine
in_cfg1_t15	96.25	$\tau 1 \to 58.03$	$\tau 1 \to 58.03$	+0.00	−35.50	refine
in_cfg1	26.07	$\tau 0 \to 49.07$	$\tau 4 \to 25.92$	−23.15	−0.19	refine
l256_cfg2	8.01	$\tau 4 \to 8.05$	$\tau 4 \to 8.05$	+0.00	+0.07	bypass
l256_cfg1_t15	78.80	$\tau 1 \to 52.27$	$\tau 1 \to 52.27$	+0.00	−25.42	refine
xl384v2_cfg2	6.70	$\tau 4 \to 6.71$	$\tau 4 \to 6.71$	+0.00	−0.06	refine
xl384v2_cfg1_t15	140.19	$\tau 1 \to 79.95$	$\tau 1 \to 79.95$	+0.00	−59.49	refine
randarv2_cfg	5.87	$\tau 4 \to 5.90$	$\tau 4 \to 5.90$	+0.00	+0.02	bypass
randarv2_t15	35.98	$\tau 1 \to 25.66$	$\tau 2 \to 24.70$	−0.96	−11.84	refine
vq_bal_t10	16.92	$\tau 4 \to 16.92$	$\tau 0.5 \to 16.82$	−0.10	−0.19	refine
vq_bal_t15	37.38	$\tau 0 \to 25.88$	$\tau 0 \to 25.88$	+0.00	−7.81	refine
lf_d0.01	76.40	$\tau 0 \to 64.84$	$\tau 1 \to 54.27$	−10.57	−21.28	refine
lf_d0.03	74.83	$\tau 0 \to 63.62$	$\tau 1 \to 52.80$	−10.82	−22.64	refine
lf_d0.1	74.86	$\tau 0 \to 63.55$	$\tau 1 \to 52.86$	−10.69	−21.81	refine

F Repair–Preservation Tradeoff Across Refiners

Figure 5 summarizes, for each refiner, how much severe-regime repair a fixed threshold retains against the FID cost it imposes on tuned samples, complementing the fixed-τ rows of Table 1 and Table 5.

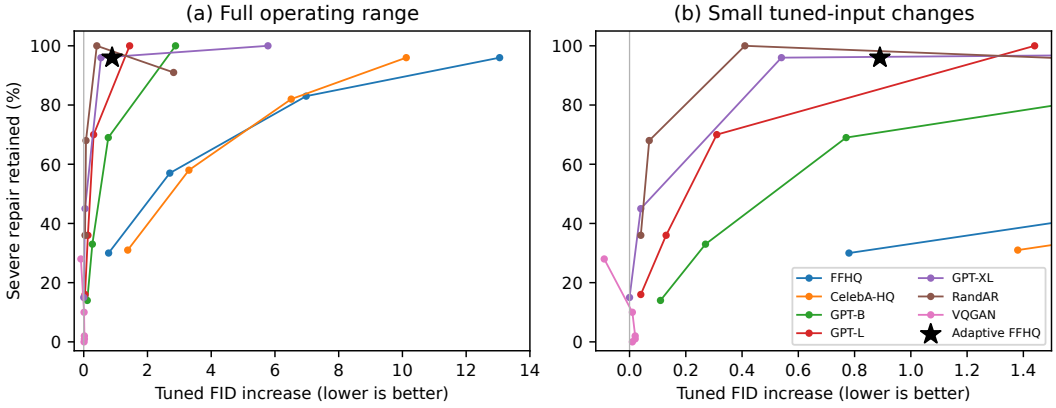

Figure 5: **Repair versus cost on tuned samples.** Horizontal position is tuned FID increase; vertical position is the reported percentage of severe-regime repair retained. Lines use $\tau \in \{1, 2, 3, 4\}$, with an additional 0.5 point for VQGAN. The repair normalization is per refiner, so percentages are not absolute cross-dataset FID gains. AGR without bypass on FFHQ retains 96% of reference repair with $+0.89$ tuned FID. This is a tradeoff, not a certified safe region.

G Inference Cost and Speculative Decoding

Table 11 measures the complete routed policy end to end on one NVIDIA RTX A5000, with the same boundary as the speculative-decoding timings below: autoregressive token generation with a KV cache in bf16 plus every refiner pass, CUDA-synchronized, VQ decoding excluded. The routing statistic on FFHQ is the mean log-probability from the unmasked pass, so detection is one refiner forward per image; flagged images then receive the full gate ($\tau = 1$, $S = 8$, 24 forwards). The mixture is $50/50$ tuned and severe within every batch; $n = 40$ images per condition at batch 1 and 100 at batch 25. On tuned input AGR adds 0.1% and 1.7% to generation time at the two batch sizes, against 2.3% and 27.6% for always-on gating; on severe input both pay the full gate; on the mixture AGR sits at the opening fraction. At batch 1 the sequential decode dominates and every refiner cost is small; at batch 25 the refiner's parallel passes are a larger share, and the closed branch saves 26 points of overhead on healthy input.

Table 11: End-to-end seconds per image for the routed policy. "Plain" is generation alone; percentages are overhead over plain; "Open" is the fraction of images the router sent to the gate. Detection costs 0.004 s (batch 1) and 0.002 s (batch 25) per image; the full gate costs 0.095 s and 0.046 s.

Batch	Input	Plain	Always-on gate, $\tau = 2$	AGR	Open (%)
1	Tuned	4.071	4.166 (+2.3%)	**4.075 (+0.1%)**	0
1	Severe	4.070	4.165 (+2.3%)	4.169 (+2.4%)	100
1	Tuned + severe	4.072	4.166 (+2.3%)	**4.123 (+1.3%)**	50
25	Tuned	0.167	0.213 (+27.6%)	**0.170 (+1.7%)**	1
25	Severe	0.167	0.213 (+27.8%)	0.215 (+29.0%)	100
25	Tuned + severe	0.167	0.213 (+27.8%)	**0.192 (+15.1%)**	49

The remainder of this appendix reports the composition of the gate with relaxed speculative decoding discussed in Section 2. We evaluate relaxed speculative decoding following LANTERN (Jang et al., 2025) with the ImageNet GPT-B decoder. At batch one, relaxed decoding plus the gate retains a 2.60–$2.64\times$ latency advantage over plain decoding. The best reported calibrated repaired FID is nevertheless 52.80, far from plain GPT-B's 8.18. At batch 25, repair raises total time above the

plain baseline. These measurements establish partial repair with a low-batch latency advantage, not quality-matched acceleration.

Table 12: **Speculative-decoding latency and quality.** Times are seconds per image at the indicated batch size. The gate uses $S = 8$; repaired FIDs use validation-selected $\tau = 1$. Target-forward counts exclude refiner forwards. Batch regimes are compared separately.

Configuration	Target fwd/image	Batch 1	Batch 25	FID
Plain GPT-B	256	5.00	0.167	8.18
Plain GPT-L	256	10.00	0.333	8.01
Exact speculation ($\delta = 0$)	215	5.26	0.518	7.96
Relaxed $\delta = 0.01$	52	1.72	0.112	76.40
+ gate	52	1.90	0.228	54.27
Relaxed $\delta = 0.03$	51	1.72	0.111	74.84
+ gate	51	1.90	0.226	52.80
Relaxed $\delta = 0.1$	51	1.72	0.108	74.86
+ gate	51	1.90	0.223	52.86

Calibration reduces relaxed-run FID from approximately 75–76 to 53–54, recovering about 22 points. Exact speculation has near-baseline FID but is slower in both measured batch regimes. Repair is not free: for $\delta = 0.03$, batch-one time rises from 1.72 to 1.90 seconds and batch-25 time from 0.111 to 0.226 seconds. The standalone gate costs 0.172 seconds per image at batch one and 0.115 at batch 100 for $S = 8$, decreasing to 0.085 and 0.058 for $S = 4$. These end-to-end timings include decoding and I/O stages, so they are not directly comparable with the per-forward counts of Appendix C.

H Additional Visual Results

Figures 6 and 7 predate adaptation and retain their original settings; they support threshold sensitivity and empirical PLL dynamics. Figure 8 plots the score-deficit relation discussed in Section 4.5, and Figure 9 shows the routed policy's outputs on an ImageNet mixture. Table 13 gives the precision and recall (Kynkäänniemi et al., 2019) behind panel (b) of Figure 6: as τ tightens, precision falls from the $\tau = 0$ maximum toward the degraded baseline while recall rises above it, and $\tau \in [0.25, 1]$ improves both over the baseline. The policy tradeoff figure in Appendix F uses the calibrated results.

Table 13: Precision, recall, Inception score, and FID along the gate's threshold sweep on severe FFHQ ($t = 1.5$, modern pipeline, 10k samples). The no-repair row is the degraded input.

Setting	Precision	Recall	IS	FID
No repair	0.071	0.486	5.19	102.71
Gate, $\tau = 0$	0.377	0.374	4.61	45.15
Gate, $\tau = 0.25$	0.379	0.391	4.70	45.68
Gate, $\tau = 0.5$	0.365	0.418	4.64	45.34
Gate, $\tau = 0.75$	0.354	0.443	4.74	46.01
Gate, $\tau = 1$	0.339	0.453	4.80	47.38
Gate, $\tau = 2$	0.269	0.509	4.93	55.00
Gate, $\tau = 4$	0.114	0.520	5.25	85.65

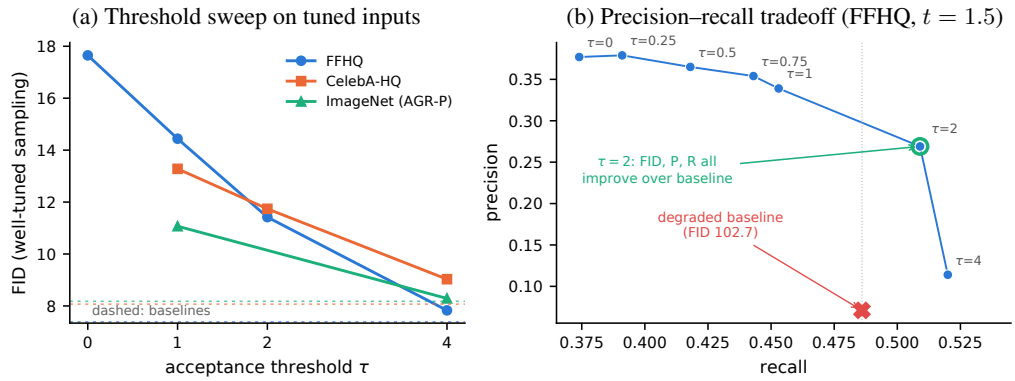

Figure 6: **Threshold-dependent quality and diversity tradeoffs.** *(a)* On well-tuned sampling, FID recovers monotonically toward the dashed baselines as τ tightens — the dose–response behind the limited editing on healthy inputs. *(b)* Under severe damage (FFHQ, t=1.5), sweeping τ traces a smooth precision–recall frontier from a precision-maximal point (τ=0) to a recall-preserving one (τ=4, recall above the degraded baseline's own); at τ=2 (circled) FID, precision, and recall all improve over the baseline simultaneously.

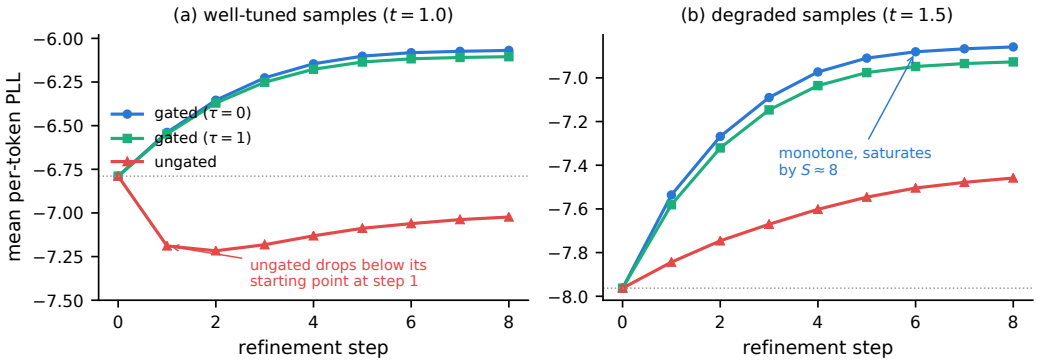

Figure 7: **Empirical PLL ascent under the gate.** Mean per-token pseudo-log-likelihood after each refinement step (FFHQ, 512 sequences; dotted line marks the starting value). Gated refinement improves at every step and saturates by $S \approx 8$ on both healthy (a) and degraded (b) inputs; ungated refinement of healthy samples falls *below its starting score after a single step* and never recovers — the self-reinforcement failure mode of Appendix A.1, measured on real data in the refiner's own objective.

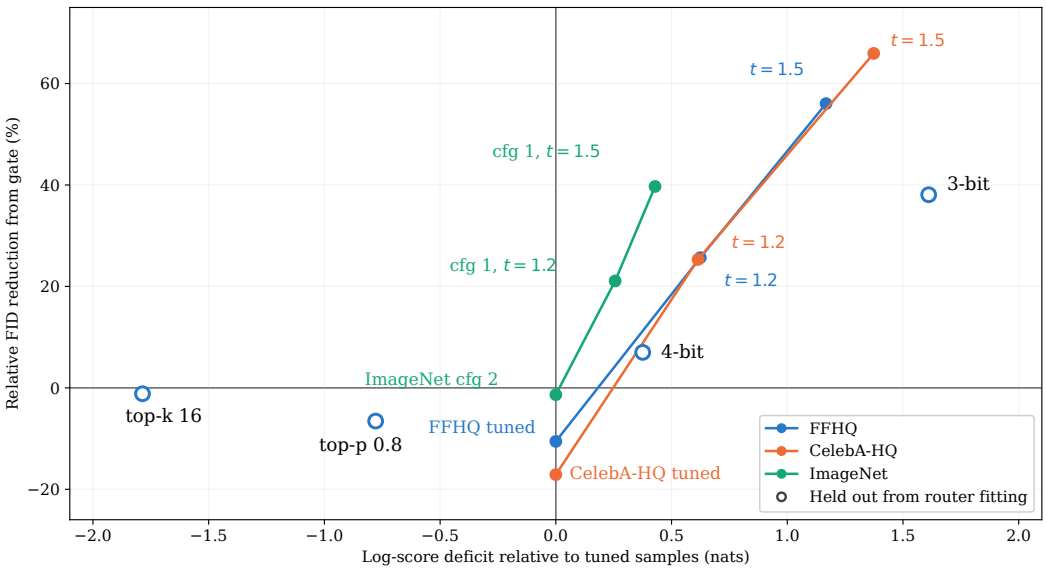

Figure 8: **Score deficit and gate gain.** Deficits use 10k-test medians of per-image unmasked mean log-score for temperature series and recorded aggregate FFHQ scores for held-out settings. Lines join temperature series; open markers denote four held-out interventions. Gain is relative FID reduction (held-out thresholds: Table 4a). ImageNet log-score and router confidence order its three plotted conditions identically. Along each temperature series, gain grows with deficit.

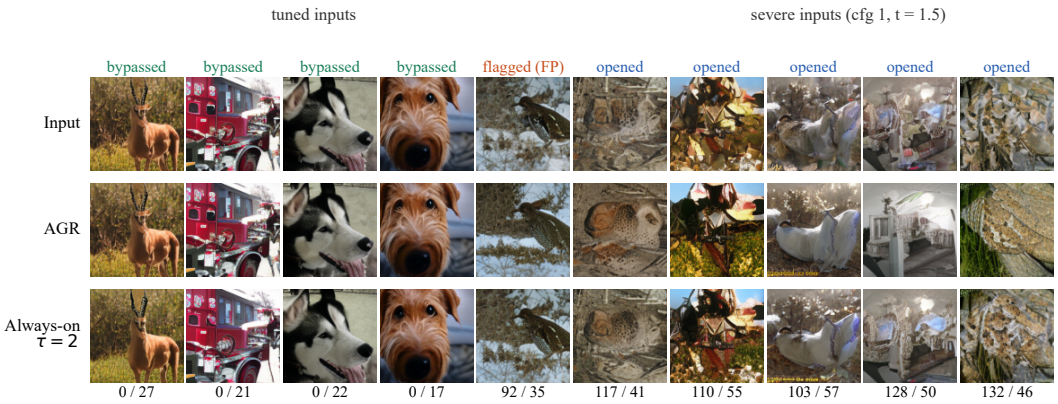

Figure 9: **AGR outputs on an ImageNet tuned/severe mixture.** Ten images chosen by the router's own decisions: four tuned images it bypassed, the tuned image it flagged most strongly (a false positive), and five severe images it opened. Rows show the input, the AGR output, and always-on gating at $\tau = 2$; numbers are tokens changed by each. Bypassed outputs are the input tokens exactly, while always-on gating rewrites 17–27 tokens of each; opened severe images have 103–132 tokens rewritten and are visibly repaired, though not all become good images.

I Supporting Legacy-Backbone Experiments

These experiments isolate sequential gating on an older face-generation pipeline. Here and in Appendices J through M, "AGR" denotes the sequential gate of Appendix A.2, the analyzable special case of the parallel gate used in the main text. They support the mechanism analysis; the modern-backbone comparisons in Section 4 provide the primary evidence for the repairability boundary. Metrics from the two protocols should not be pooled.

Legacy image setting. We use FFHQ (Karras et al., 2019) and CelebA-HQ (Karras et al., 2018) at 256×256. The same pretrained FFHQ transformer from (Esser et al., 2021) (307M parameters,

Table 14: Supporting legacy results on FFHQ and CelebA-HQ (256×256, $50{,}000$ generated images per dataset). Our AR baseline uses the released FFHQ checkpoint of (Esser et al., 2021) and the sampling parameters reported there (k=300, T=1.0); the same checkpoint and refiner are used for both datasets without any dataset-specific retraining. FID is computed against the $70{,}000$-image FFHQ training set for the FFHQ column and against the $30{,}000$-image CelebA-HQ set (Karras et al., 2018) for the CelebA-HQ column, using a single fixed pipeline reused across every frozen-backbone row below. The gap between our AR baseline (15.36 FFHQ) and the 9.6 originally reported in (Esser et al., 2021) likely reflects evaluation-pipeline differences (FID implementation, reference set, hardware) that we could not reconcile from public materials, so we report only what we can reproduce; all within-table comparisons share our pipeline and are therefore unaffected. The method ordering is preserved across both datasets.

Method	**FFHQ FID** $\downarrow$	**CelebA-HQ FID** $\downarrow$
Reference numbers from prior work (different evaluation pipelines)		
VDVAE (Child, 2021)	28.5	—
VQGAN + Transformer (Esser et al., 2021)	9.6	10.2
MaskGIT (Chang et al., 2022)	7.8	—
Frozen-backbone correction, our pipeline (ours and baselines)		
AR baseline	15.36	18.14
+ Argmax resampling	11.68	16.04
+ Ungated resampling	10.91	14.78
+ AGR (τ=2.0)	10.53	13.66
+ AGR (τ=1.0)	9.85	13.47
+ **AGR (**τ=0.0**, ours)**	**9.48**	**13.40**

VQGAN codebook size 1024, sequence length 256) serves as the AR backbone for both datasets; it is *never modified*, and baseline samples use its original top-k protocol (k=300, T=1.0). The refiner is a bidirectional transformer (d=512, 8 layers, 8 heads, 26.4M params, vocab 1025 incl. mask) trained for 400k steps by masked denoising on FFHQ tokens from the same VQGAN; both backbone and refiner are applied to CelebA-HQ without retraining, isolating the refinement procedure as the only variable across datasets (Appendix K). We evaluate FID (Heusel et al., 2017) on $50{,}000$ generated images per dataset, against the $70{,}000$-image FFHQ training set (FFHQ rows) and the $30{,}000$-image CelebA-HQ set (Karras et al., 2018) (CelebA-HQ rows), and report the refiner's pseudo-log-likelihood (Eq. (6), nats summed over 256 positions) before and after refinement.

Legacy methods compared. We compare four test-time procedures: *(i)* **No refinement** (raw AR sample); *(ii)* **Ungated resampling** — identify the k least-confident positions and commit a sample from the refiner distribution at each (the $\tau \to -\infty$ limit of AGR; not MaskGIT, which generates from a fully-masked start); *(iii)* **Argmax resampling** — as above, but commit the refiner's argmax (the greedy variant FM1 warns against); *(iv)* **AGR (ours)**: the sequential gate of Appendix A.2 with $\tau \geq 0$. All refinement methods share a 4-step cosine schedule ($\rho_{\max}$=0.30, $\rho_{\min}$=0.05) and proposal temperature β=1.0; the only intended difference is the selection/acceptance rule. The AGR threshold is τ=0.0 unless otherwise stated (Appendix I.3 sweeps it). We do not claim state-of-the-art generation; the relevant comparison is within a fixed frozen-backbone pipeline.

I.1 Correction Results

Within this fixed pipeline, sequential gating reduces FFHQ FID from 15.36 to 9.48 and CelebA-HQ FID from 18.14 to 13.40. The ungated-to-gated gaps are 1.43 and 1.38 FID, respectively. The same ordering occurs on both datasets, although they share a backbone and refiner and are not independent tests of broad transfer. Prior-work FIDs use different evaluation pipelines and are contextual references, not matched comparisons.

Two features of this table differ from the modern results and are explained by the score-deficit rule of Section 4.5. First, ungated and argmax resampling *improve* the legacy baseline here, whereas ungated refinement damages tuned modern samples (Table 1). The legacy baseline is weak (FID 15.36 against the 9.6 reported for the same checkpoint), so its samples sit well below the refiner's typical score, the analogue of a degraded input; in that regime even unselected edits help, as they do on severe



Figure 10: **Qualitative comparison on CelebA-HQ.** Five CelebA-HQ samples (rows) rendered under all six methods (columns): frozen AR baseline, ungated stochastic resampling (*Plain*), AGR at acceptance thresholds $\tau \in \{0, 1, 2\}$, and greedy argmax resampling. AR seeds are shared across columns within each row, so differences reflect the refinement procedure rather than sampling noise. AGR repairs local structural inconsistencies (eye asymmetries, hair discontinuities, skin-region artifacts) in some displayed samples. Appearance can also change; these selected pairs do not establish identity preservation or a population-level success rate.

ImageNet in Table 1, and the gate's contribution is the further 1.4 FID over ungated editing at half the token changes (Table 15). Second, the best legacy threshold is $\tau = 0$ and larger thresholds hurt, whereas tuned modern samples prefer the strictest gate or bypass; this is the same relation between deficit and useful edit volume seen in Figure 8. The legacy ungated baseline is also sequential with $\rho_{\max} = 0.30$ over $S = 4$ steps, less aggressive than the parallel MaskGIT-style baseline used on modern backbones. Recall (Kynkäänniemi et al., 2019) increases from 0.502 to 0.557 on FFHQ and from 0.379 to 0.439 on CelebA-HQ. The paired grids in Appendix M illustrate local edits without establishing identity preservation or a per-image success rate.

I.2 Empirical PLL Improvement on In-Domain Samples

On $50{,}000$ FFHQ sequences, sequential gating at $\tau = 0$ raises mean PLL from -1165.17 to -1064.17 nats, with an end-to-end increase for every tested sequence. Ungated resampling raises mean PLL by 52 nats but lowers it for 6.9% of sequences. This is an empirical observation for this checkpoint and protocol, not a proof of stepwise monotonicity or generalization to another distribution. The fraction of sequences with lower PLL does not quantitatively explain an aggregate FID difference; the no-CFG experiment further demonstrates that PLL and image quality can move in opposite directions.

Table 15: FFHQ ablation over the acceptance threshold τ. "Accept rate" is the fraction of proposals accepted; "Avg accepted Δ" is the mean log-prob gain over accepted edits only.

Condition	Accept rate	Change rate	Avg accepted Δ	FID $\downarrow$
No refinement	—	0.000	—	15.36
Argmax resampling	1.000	0.867	—	11.68
Ungated resample ($\tau \to -\infty$)	1.000	0.724	—	10.91
AGR ($\tau = 0.0$)	0.514	0.361	1.36	**9.48**
AGR ($\tau = 1.0$)	0.296	0.208	2.19	9.85
AGR ($\tau = 2.0$)	0.136	0.096	2.99	10.53

I.3 Ablation: Isolating the Acceptance Gate

Removing the sequential gate increases FID from 9.48 to 10.91 in this legacy setting. Among the tested nonnegative thresholds, $\tau = 0$ has the lowest FID. Raising τ from 0 to 2 reduces acceptance from 51.4% to 13.6% and final token changes from 36.1% to 9.6%, while increasing the mean accepted score gain from 1.36 to 2.99 nats. FID nevertheless worsens to 10.53: larger accepted score gains are not equivalent to larger quality gains. This tradeoff contrasts with the tuned modern controls, where a stricter gate limits degradation. It does not support a universal choice of τ. The legacy protocol uses $S = 4$ and $\beta = 1$; in that setting FID is non-monotone in S with its minimum at $S = 4$, and $\beta \in [0.8, 1.2]$ changes FID by under 0.7. The modern protocol instead uses $S = 8$ with the parallel gate (Section 4.1).

Inference cost. At our settings (S=4, T=256, $\rho_{\max}$=0.30), AGR runs at 3.426 s/image versus 0.584 s for the AR baseline on a single Tesla P40 ($\approx 487\%$ overhead), placing it in the same regime as test-time compute scaling for language models (Snell et al., 2024).

J Structured Rule Distribution Details

The structured rule distribution used in toy experiments has the following parameters:

- Vocabulary: $|\mathcal{V}| = 64$, split into even vocab $\{0, \ldots, 31\}$ and odd vocab $\{32, \ldots, 63\}$.
- Sequence length: $T = 32$.
- Anchor tokens: $\{4, 5, 6\}$ for positions $0, \ldots, 15$; $\{36, 37, 38\}$ for positions $16, \ldots, 31$.
- Forbidden bigrams: 10 pairs, listed in the code release.

Dataset size: 10,000 training sequences, 2,000 evaluation sequences. All sequences are sampled to satisfy all four constraints exactly.

Toy setting. We construct a structured discrete distribution over vocabulary $|\mathcal{V}| = 64$ and length $T = 32$ with four simultaneous constraints: positional parity (even positions draw from $\{0, \ldots, 31\}$, odd from $\{32, \ldots, 63\}$), no consecutive repeats ($x_t \neq x_{t-1}$), anchor coverage (each half contains at least one designated anchor token), and forbidden bigrams (a fixed list). The constraints are learnable by a bidirectional transformer but hard enough that a weak causal AR transformer ($d = 128$, 4 layers) sampled at temperature 1.5, top-k 16 produces systematic structured errors. The refiner is a bidirectional transformer of matched size ($d = 128$, 4 layers) trained by masked denoising on sequences from the same distribution. Architecture details are in Appendix K. We evaluate on $2{,}000$ AR-generated sequences (primary benchmark) and on $2{,}000$ sequences with 30% random token corruption (control), reporting rule accuracy (fraction of satisfied constraints per sequence), sequence accuracy (fraction of fully valid sequences), and change rate (fraction of tokens modified).

J.1 Toy Validation

The toy benchmark provides ground-truth rule satisfaction unavailable on real images, isolating the gate's contribution from sampling noise. On AR-generated structured errors, AGR (τ=0) raises rule accuracy to 0.963 and changes only 37.5% of tokens, while argmax and ungated resampling collapse sequence accuracy to ≤ 0.001 at change rates of 72–87% — a clean confirmation of FM1 in a setting

where the failure mode is not masked by softer real-image constraints. Full experimental results (illustrated by Table 16 and Figure 11) and a discussion of why sequence accuracy can drop while rule accuracy rises (a distinction between local and sequence-level objectives) appear in Appendix J.2.

J.2 TOY VALIDATION RESULTS

This subsection contains the full toy-validation results summarized in Appendix J.1.

Table 16: Toy validation, 2,000 evaluation sequences per condition. Rule acc. is fraction of constraints satisfied; Seq. acc. is fraction of fully valid sequences; Change rate is fraction of tokens modified.

	AR-generated errors (primary)			Random corruption (control)		
Condition	Rule acc. ↑	Seq. acc. ↑	Change rate	Rule acc. ↑	Seq. acc. ↑	Change rate
No refinement	0.946	0.877	0.000	0.822	0.630	0.000
Ungated resample	0.725	0.001	0.724	0.683	0.001	0.733
Argmax resample	0.619	0.000	0.867	0.621	0.000	0.874
AGR (τ=0, ours)	**0.963**	**0.848**	0.375	**0.897**	**0.733**	0.378

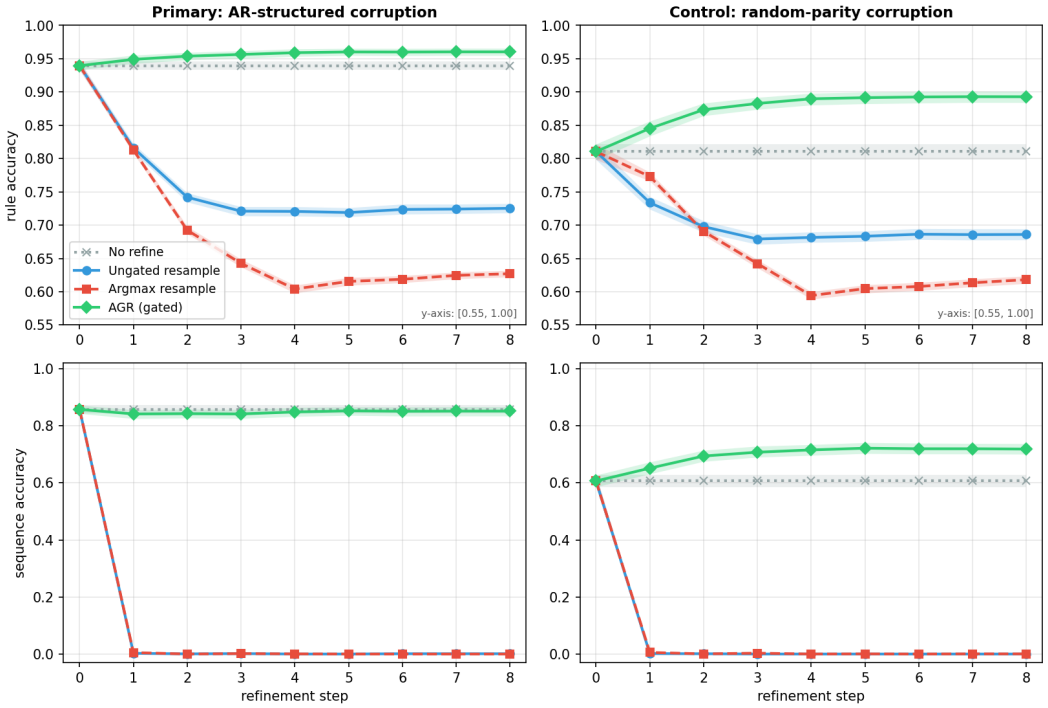

Figure 11: **Refinement step curves in the toy setting.** Per-step rule accuracy (top, zoomed) and sequence accuracy (bottom) over 2,000 sequences; shaded bands are 95% bootstrap CIs. All methods share a MaskGIT-style cosine position schedule; only the log-probability acceptance gate differs. AGR lifts rule accuracy from 0.94→0.96 on primary ($\sim$30% of the remaining error) and by 8/11 rule/seq points on control; ungated and argmax collapse sharply (FM1).

On AR-generated structured errors, AGR improves rule accuracy by +0.238 and sequence accuracy by +0.847 over ungated resampling, while changing only 37.5% of tokens compared to ungated's 72.4%. The acceptance gate's selectivity — accepting roughly 24% of proposals — is the mechanism: AGR commits only to edits that strictly improve the refiner's conditional score, avoiding the self-reinforcement dynamic of FM1. Argmax resampling uniformly degrades quality across both error types and all metrics, directly confirming FM1: the self-reinforcing overconfidence loop destroys sequence validity even from high-quality starting points. On random corruption (control), AGR outperforms all baselines but by a smaller margin over no-refinement, consistent with AGR's intent to correct structured AR errors rather than reconstruct heavily corrupted inputs.

Why sequence accuracy can drop while rule accuracy rises. On the primary benchmark, AGR raises rule accuracy from 0.946 to $\mathbf{0.963}$ but lowers sequence accuracy from 0.877 to 0.848. This is consistent with AGR's local-conditional guarantee: every accepted edit improves the refiner's score *at the edited position* (Proposition 1), but the toy distribution's hard cross-position constraints (forbidden bigrams, no consecutive repeats) can be violated by an edit that the refiner locally prefers. A single such edit converts a fully valid sequence into a sequence with one rule violation, which counts as a $+1/4$ change in rule accuracy (positive on average) but flips the sequence-accuracy bit from 1 to 0. AGR therefore improves the average constraint-satisfaction rate while occasionally making a globally valid sequence locally inconsistent. On real images, sequence-level validity is not directly measured. Appendix I.2 reports a PLL increase for all tested legacy FFHQ sequences, but PLL is neither sequence validity nor image quality, so that observation cannot exclude an analogous quality regression on images.

K Model Architectures

Toy AR model. Causal transformer: $d = 128$, 4 layers, 4 attention heads, MLP ratio 4, GELU activation, LayerNorm pre-norm. Trained for 3000 steps, batch size 128, AdamW (lr $= 3 \times 10^{-4}$, weight decay 0.1), gradient clipping 1.0.

Toy refiner. Bidirectional transformer: $d = 128$, 4 layers, 4 attention heads, otherwise identical to the AR model. Trained for 4000 steps with uniform mask-ratio schedule $\rho \in [0.05, 0.50]$.

Legacy image refiner. Bidirectional transformer: $d = 512$, 8 layers, 8 heads, 26.4M parameters. Vocabulary size 1025 (1024 codebook tokens + MASK token at index 1024). Trained for 400k steps on FFHQ token sequences with uniform mask-ratio schedule $\rho \in [0.05, 0.50]$, batch size 64 across $4\times$ Tesla P40 GPUs, AdamW (lr $= 3 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay 10^{-2}), cosine LR decay with 2000-step linear warmup.

Modern-backbone refiners. All modern refiners are bidirectional transformers over the LlamaGen VQ-16 vocabulary (16,384 codes plus `[MASK]`), trained by masked denoising with the mask ratio drawn uniformly from $[0.05, 0.5]$ per sequence and cross-entropy on the selected positions; the AR generator is frozen throughout and the last checkpoint is used. Table 17 lists the configurations. The face refiners use AdamW with learning rate 3×10^{-4}, weight decay 0.01, and a 2k-step warmup on horizontally flip-augmented token sequences. The ImageNet refiners are class-conditional with 10% label dropout, corrupt selected positions with an 80/10/10 mask/random/keep split, train in bf16, and keep an EMA (decay 0.9995) that is used at inference; they train at roughly 100 sequences per second on one NVIDIA A5000 (about 13 and 20 GPU-hours). Held-out masked accuracy was logged but not used for checkpoint selection.

Table 17: Modern-backbone refiner training configurations.

Refiner	Params	Architecture	Training sequences	Steps × batch
FFHQ	42.1M	8 layers, $d = 512$, 8 heads	70,000	400k × 16
CelebA-HQ	42.1M	8 layers, $d = 512$, 8 heads	56,000	400k × 16
ImageNet 256^2	155M	12 layers, $d = 768$, 12 heads	120,500	150k × 32
ImageNet 384^2	156M	12 layers, $d = 768$, 12 heads, $T = 576$	251,240	300k × 24

L CelebA-HQ Gate Behavior

This appendix complements the FFHQ pseudo-log-likelihood validation in Appendix I.2 and the qualitative comparison in the paired grids of Appendix M with two CelebA-HQ-specific visualizations of AGR's gating behavior: the empirical distribution of per-proposal acceptance scores Δ_i, and the spatial distribution of accepted edits over the token grid. Both are computed on a fixed CelebA-HQ probe set of 16 images sampled with the AR backbone under the protocol of Appendix I (k=300, temperature 1.0) and refined for S=4 steps under the cosine mask-ratio schedule ($\rho_{\max}$=0.30, $\rho_{\min}$=0.05), yielding a total of n=8,861 single-token proposals. The probe set is reproducible from the seed reported with the released code.

L.1 PER-PROPOSAL ACCEPTANCE DISTRIBUTION

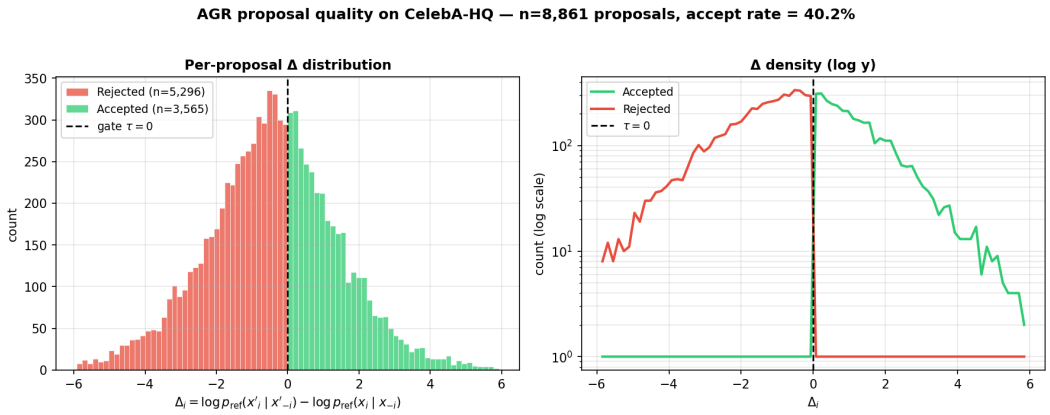

Figure 12: **Distribution of per-proposal acceptance scores Δ_i on CelebA-HQ.** Linear-scale histogram (left) and log-scale density (right) over all n=8,861 proposals generated during refinement of the CelebA-HQ probe set. Accepted proposals ($\Delta_i > \tau$=0, green) and rejected proposals ($\Delta_i \leq 0$, red) are cleanly separated by the gate. The accept rate is 40.2%, lower than the 51.4% observed on FFHQ at the same τ=0 (Table 15). The log-scale panel shows that the rejected tail extends down to $\Delta_i \approx -6$ nats, quantifying the magnitude of harmful edits the gate suppresses — exactly the failure mode that the 6.9% PLL-degradation rate of ungated resampling (Appendix I.2) captures on FFHQ.

Figure 12 provides a CelebA-HQ analogue of the FFHQ PLL validation in Appendix I.2. Whereas the FFHQ result reports the aggregate effect of the gate (100% of $50{,}000$ sequences improve in PLL), the histogram visualizes the per-proposal quantity that the gate actually filters on. Two observations are worth noting. First, both the accepted and rejected distributions place substantial mass close to the gate boundary τ=0, so the choice of τ is empirically meaningful rather than vacuous and is consistent with the FFHQ τ-sweep behavior in Table 15. Second, the rejected distribution is roughly centered around $\Delta_i \approx -1$ nat, so on average the gate is rejecting edits that would lower the refiner's local conditional log-probability by a non-trivial amount — not just edits at the noise floor.

L.2 SPATIAL PATTERN OF ACCEPTED EDITS

Figure 13 extends the qualitative comparison in the paired grids of Appendix M with a spatial map of *where* the gate accepts edits. The qualitative grids of Appendices I and M show the *result* of refinement (final pixel output); this figure instead shows the *distribution of accepted edits over the token grid*, which is the quantity that distinguishes AGR from a uniform per-token refinement schedule. The visible concentration of accepts on facial features and structural boundaries is consistent with the design of AGR: the refiner's local conditional score $s_i(\cdot; x_{\setminus i})$ is most peaked — and therefore most likely to produce $\Delta_i > 0$ — where the bidirectional context provides strong constraints, which is typically exactly these structural regions. Background and uniform-color regions receive few accepts because either the original AR token is already close to the refiner's argmax (small $|\Delta_i|$) or the proposal is rejected as not strictly improving.

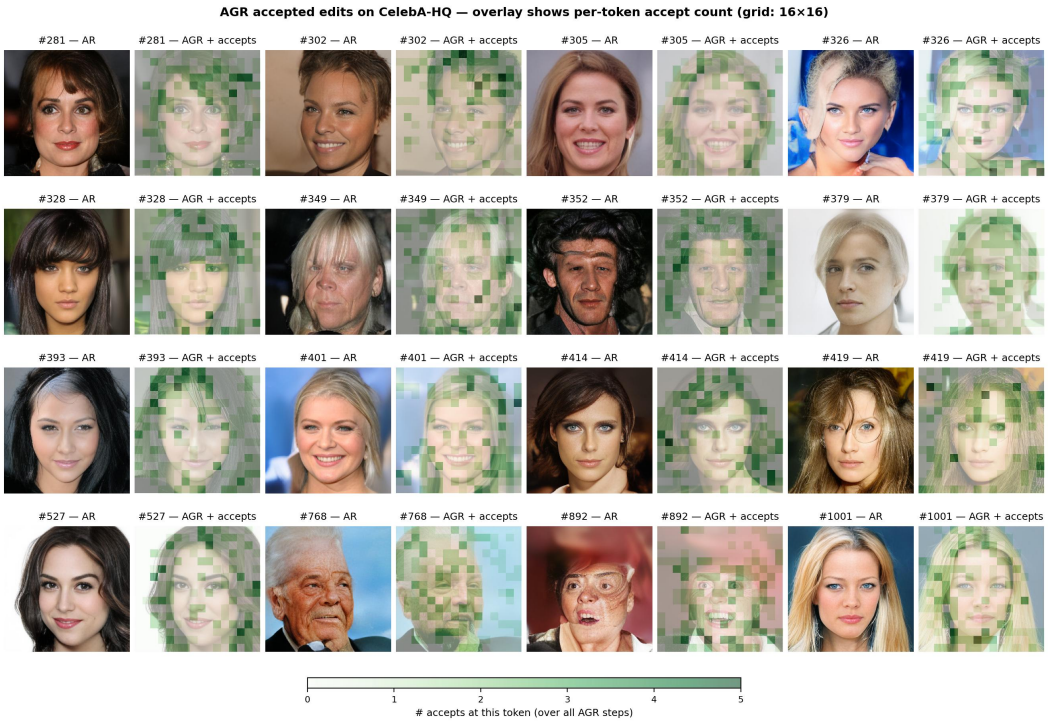

Figure 13: **Spatial acceptance pattern on CelebA-HQ.** For 16 images from the CelebA-HQ probe set, each pair shows the original AR sample (left) and the same AR sample with a per-token accept-count heatmap overlaid on the 16×16 token grid (right). The colorbar at each token position counts how many of the $S{=}4$ refinement steps accepted an edit at that position (range 0–4). Sample IDs (#281, #302, . . .) index the probe set described in Appendix L. Acceptance is concentrated around facial structure (eyes, lip contour, hairline) and along background-to-foreground transitions, rather than uniformly across the grid.

M Additional Qualitative Comparisons on FFHQ

This appendix complements the CelebA-HQ qualitative comparison of Figure 10 with a larger set of FFHQ samples, the dataset on which the sequential gate delivers its legacy FID improvement ($15.36 \rightarrow 9.48$, Appendix I.1). Each row shows a single AR sequence rendered under all six methods evaluated in Table 14: the frozen AR baseline (top-$k{=}300$, $T{=}1.0$), ungated stochastic resampling (*Plain*), AGR at three acceptance thresholds $\tau \in \{0, 1, 2\}$, and greedy argmax resampling. Method ordering and refinement schedule are identical to the legacy experiments ($S{=}4$, $\rho_{\max}{=}0.30$, $\rho_{\min}{=}0.05$). Samples were selected without cherry-picking for any specific method; AR seeds are shared across columns within a row, so differences are attributable to the refinement procedure rather than to sampling noise.

Across the five samples in Figure 14, the same trends visible on CelebA-HQ recur: AGR repairs local structural inconsistencies (eye and ear asymmetries, hair discontinuities, glasses artifacts, fine skin-region noise) in some examples, but neither these images nor the reported metrics establish identity preservation. Argmax resampling also produces visible changes and smoothing in these examples. Plain (ungated) resampling occasionally helps but also introduces visible artifacts on samples where the AR baseline is already plausible — the failure mode the gate is designed to suppress.

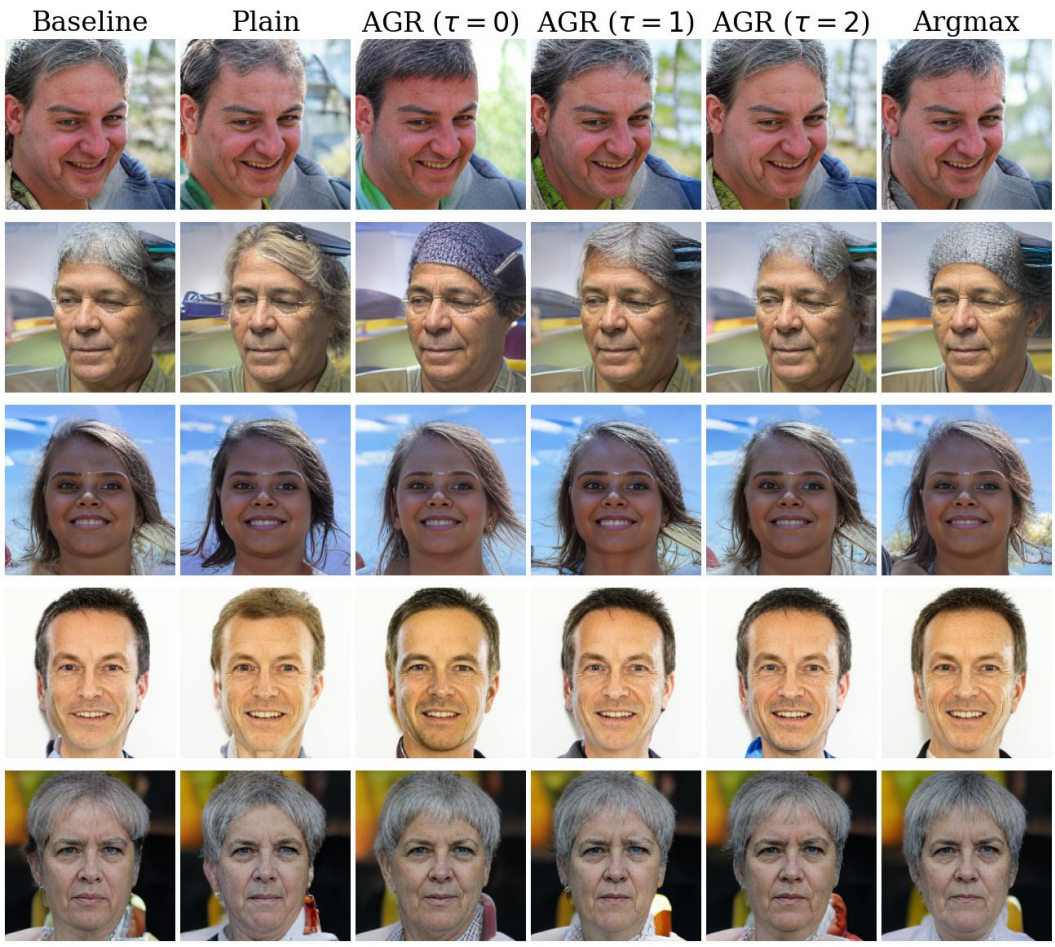

Figure 14: **Qualitative comparison on FFHQ.** Five FFHQ samples under all six methods. Columns: frozen AR baseline, ungated stochastic resampling (*Plain*), AGR at $\tau \in \{0, 1, 2\}$, and argmax resampling.