SELFVERIFY: TYPICALITY-GUIDED SAMPLING WITH DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models are trained to predict noise under a prescribed Gaussian corruption process, suggesting that their own noise predictions may provide an internal signal for candidate selection. We introduce SELFVERIFY, which compares patchwise mean and variance statistics of predicted noise with a Gaussian reference and retains the lowest-scoring trajectory. SELFVERIFY requires neither a separately trained external reward or preference model nor additional denoiser evaluations, avoiding reliance on external-verifier training and preference data for selection. We interpret the resulting score as a noise-statistical typicality proxy, with preference and alignment gains treated only as downstream empirical outcomes. Controlled noise-process interventions show that predicted-noise statistics respond to selected departures from the training process, including variance changes missed by a patch-mean statistic. On SANA-base, SELFVERIFY improves several preference metrics under standard classifier-free guidance. Motivated by the possibility that guidance-induced off-manifold behavior may weaken this signal, we additionally evaluate SELFVERIFY under CFG++, an independently proposed manifold-constrained guidance method, and observe larger relative selection gains. A scored-branch control further shows that the prediction used for ranking materially affects performance. Gains on GenEval2 and AudioLDM2 further suggest that the signal transfers across tasks and modalities. Overall, predicted-noise typicality provides a useful internal sampling signal without establishing formal data typicality, manifold proximity, or a causal link from typicality to preference.

1 INTRODUCTION

Diffusion models generate images and audio by transforming noise into structured outputs (Ho et al., 2020; Rombach et al., 2022; Podell et al., 2023; Kong et al., 2021). Selecting among several trajectories can improve generation without retraining, but common inference-time selection methods rely on an external evaluator (Ma et al., 2025; Fernandes et al., 2025; Xie et al., 2025). We ask whether a diffusion model's own noise prediction can supply a useful selection signal before generation finishes.

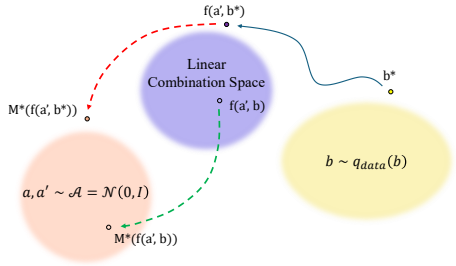

Figure 1: Conceptual figure of diffusion noise prediction as a distribution estimation. The image $b = x_0$ and Gaussian noise $a' = \epsilon$ form the noisy image $f(a', b) = x_t$; the learned model M estimates the added noise. The point b^* illustrates an image outside the training data distribution, for which the same estimator may produce different statistics.

Our starting point is the model's training distribution. Noise prediction training couples in-distribution data with a specified Gaussian noise process. Inputs that depart from those conditions can induce different predicted-noise statistics. This motivates a *noise-statistical typicality proxy* based on the local mean and variance of the model's noise estimate relative to a fixed Gaussian reference. The proxy is not a preference score. Its usefulness for preference and perceptual quality is an empirical downstream question, rather than a consequence of the noise reference alone.

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.

SELFVERIFY ranks candidate trajectories using patchwise moments of $\hat{\epsilon}$ and retains the candidate with the lowest Gaussian-reference score. The score combines three fixed, equally weighted discrepancies: squared patch mean, Gaussian KL divergence, and squared Wasserstein distance. For models that predict velocities or other parameterizations, we first recover the corresponding noise estimate using the sampler's coordinates. Selection uses neither a trained reward model nor an additional denoiser evaluation. Controlled forward-noise interventions provide a diagnostic foundation: mean and variance shifts, as well as some higher-order input shifts after passage through the model, become visible in predicted-noise moments. In particular, variance-sensitive statistics detect interventions that a patch-mean statistic can miss. These experiments establish statistical sensitivity under the tested interventions, without assuming that every OOD input must increase divergence.

The main generation study focuses on base diffusion models with classifier-free guidance (CFG) (Ho & Salimans, 2022). This setting exposes both the sampling update and the conditional and unconditional predictions, allowing us to study when the proxy is useful. SELFVERIFY improves SANA-base (Xie et al., 2025) under standard CFG, with substantially larger relative gains under our CFG++ implementation. CFG++ motivates manifold-constrained guidance (Chung et al., 2024); the stronger selection gains under that sampler are consistent with greater compatibility between its guidance dynamics and the noise-reference score. An additional control keeps the standard CFG sampler fixed while changing only the prediction branch used for scoring. These experiments separate guidance compatibility from any stronger claim that SELFVERIFY directly measures distance to a data manifold.

Beyond these guidance comparisons, GenEval2 shows that the gains extend to compositional generation. On SANA-base under CFG++, SELFVERIFY improves both Soft-TIFA and HPSv2 across the reported patch scales, while SDXL exhibits smaller and more patch-dependent effects. A SANA-base candidate-count study further shows progressively larger gains as the candidate pool increases from two to sixteen at an intermediate selection step. An AudioLDM2 (Liu et al., 2024) speech experiment extends the same selection principle beyond images, improving naturalness as measured by UTMOS (Saeki et al., 2022). In contrast, evaluations on few-step distilled samplers are mixed, marking a boundary of the current evidence rather than establishing general transfer to distilled sampling regimes. Together, these results position predicted-noise typicality as a practical, guidance-dependent signal for reward-free inference-time selection, with demonstrated gains in generation quality, compositional alignment, and speech naturalness in the evaluated settings.

Our contributions are:

- A predicted-noise typicality proxy motivated by diffusion training and examined through controlled noise-process interventions.
- A three-component, reward-free trajectory selector based on patchwise predicted-noise moments, with explicit noise conversion and guidance-branch definitions, evaluated primarily on base models under CFG and CFG++.
- A study of guidance compatibility, compositional alignment, candidate-count scaling, and transfer to speech, together with a boundary evaluation on few-step distilled samplers.

2 RELATED WORK

Diffusion parameterizations, guidance, and predicted-noise diagnostics. Diffusion models learn to reverse a prescribed Gaussian corruption process (Sohl-Dickstein et al., 2015; Ho et al., 2020), with modern variants using latent representations, transformer backbones, accelerated solvers, and few-step distillation (Rombach et al., 2022; Peebles & Xie, 2023; Esser et al., 2024; Song et al., 2020; Lu et al., 2022; Song et al., 2023; Luo et al., 2023; Sauer et al., 2024; Chen et al., 2025). Their prediction targets are not universal: models may output noise, clean samples, flow velocities, or scheduler-specific quantities (Liu et al., 2022; 2023; Lu & Song, 2025). We therefore convert every model output to a noise estimate using the corresponding sampler coordinates before scoring; Appendix C gives the exact transformations and scale checks.

Guidance further changes the prediction encountered along the sampling trajectory. CFG extrapolates conditional and unconditional predictions, whereas CFG++ constructs the clean estimate by interpolation and re-noises with the unconditional prediction from a manifold-constrained perspective (Dhariwal & Nichol, 2021; Ho & Salimans, 2022; Karras et al., 2024; Chung et al., 2024).

SELFVERIFY is not a guidance method: we compare standard CFG, an implemented CFG++ rule, and a fixed-sampler scored-branch control to study when Gaussian-referenced predicted-noise ranking is compatible with the sampling dynamics.

Predicted-noise statistics have also been used in diffusion-based anomaly and OOD analysis through reconstruction, density- or score-based quantities, and trajectory behavior (Bercea et al., 2023; Zhang et al., 2025; Yao et al., 2024; Livernoche et al., 2023; Heng et al., 2024; Aithal et al., 2024). Our diagnostic has a narrower role: it asks whether controlled changes to the forward-noise process become visible in the moments of a fixed model's predicted noise. It motivates the statistical object used for candidate ranking rather than a general-purpose OOD detector. Likewise, stronger gains under CFG++ are interpreted as evidence of compatibility between the score and sampling dynamics, not as a direct measurement of manifold distance. Appendix E further shows that clean-calibrated discrimination of natural-image shifts is substantially weaker and shift-dependent.

Inference-time selection and preference-based methods. Generative models can be improved either by training against preference or reward signals, as in diffusion preference optimization and reinforcement-learning approaches (Wallace et al., 2024; Rafailov et al., 2023; Liu et al., 2025), or by allocating additional computation at inference. Best-of-N and structured search methods select among complete or intermediate candidates using external reward, preference, or semantic models (Ma et al., 2025; Fernandes et al., 2025; Xie et al., 2025). SELFVERIFY belongs to the latter broad family in that it evaluates multiple trajectories, but differs in using predicted-noise statistics already produced by the diffusion model and making the selection before completion, without training or querying an external evaluator. We use ImageReward-based Best-of-N as an external-selector reference and report downstream preference and perceptual metrics separately (Wu et al., 2023; Xu et al., 2023; Hessel et al., 2022).

Noise statistics and controlled shifts. Real acquisition noise provides examples of deviations from a fixed homoscedastic Gaussian reference. Poissonian–Gaussian models capture signal-dependent variance in raw images (Foi et al., 2008); smartphone data show substantial variation across sensors and capture conditions (Abdelhamed et al., 2018). Quantile analyses find tails heavier than conventional Poisson/Gaussian models predict (Zhang et al., 2015), and physics-based low-light models use flexible read-noise distributions (Wei et al., 2020). These observations motivate controlled tests of mean and variance changes, together with correlation and tail shifts that preserve selected input moments. Our interventions remain isolation controls: forward diffusion noise and physical sensor noise enter different processes, and exact patch-mean preservation or moment-matched cubic noise is not a claim about a calibrated camera. This distinction lets us test what information becomes available to the retained moment-based scores, including indirect sensitivity after the fixed model maps higher-order input shifts to predicted-noise moments, without conflating that test with realism.

3 NOISE STATISTICS AS A TYPICALITY PROXY

3.1 A CONDITIONAL-ESTIMATION VIEW

Consider independent components $a \sim \mathcal{N}(0, I)$ and $b \sim q_{\text{data}}$, mixed as $x = f(a, b) = c_1 a + c_2 b$. A model trained to recover a by squared error has the ideal predictor

$$M^* = \arg\min_M \mathbb{E}\|a - M(f(a,b))\|^2, \qquad M^*(x) = \mathbb{E}_{\text{train}}[a \mid x]. \tag{1}$$

At a diffusion timestep t, this is precisely the noisy-image-to-noise prediction problem:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \qquad \hat{\epsilon} = \epsilon_\theta(x_t, t). \tag{2}$$

The correspondence is $a = \epsilon$, $b = x_0$, $c_1 = \sqrt{1 - \bar{\alpha}_t}$, $c_2 = \sqrt{\bar{\alpha}_t}$, and $M = \epsilon_\theta(\cdot, t)$. The model learns to infer the Gaussian noise from the noised image under the training data distribution. An image outside that distribution changes the input condition under which the same learned estimator is applied; its predicted-noise statistics may therefore change.

This connection applies to more than image replacement. At fixed x_0, changing the mean, variance, correlation, or tail behavior of the injected noise also changes the joint input law seen by the

learned estimator. Conversely, an ordinary sampling trajectory need not be labeled ID or OOD: our generation rule simply asks which candidate's current prediction agrees more closely with the reference statistics. The diagnostic and the selector therefore share a statistical object—the predicted noise—but answer different questions.

Figure 1 motivates inspecting predicted-noise statistics. It does not imply that an ID noise prediction is itself standard-normal: a conditional mean need not share the marginal distribution of the variable it estimates. We therefore use agreement with a Gaussian reference as an operational *typicality proxy*, and evaluate its usefulness empirically. The full DDPM correspondence, output-moment derivation, and a counterexample to universal ID/OOD ordering are given in Appendix D.

One useful identity makes this limitation explicit. If the squared-error optimum is attained, then

$$\mathbb{E}\|\epsilon\|^2 = \mathbb{E}\|\mathbb{E}[\epsilon \mid x_t, t]\|^2 + \mathbb{E}\,\mathrm{tr}\,\mathrm{Var}(\epsilon \mid x_t, t). \tag{3}$$

Thus the predicted noise can have a second moment below that of the injected noise even on training-distribution inputs. Our reference is a fixed anchor for comparing candidates under the same model, timestep, and guidance setting; it is not a claim of perfect marginal calibration.

3.2 Moment-Based Reference Scores

For a predicted-noise patch containing M elements, let $\mu = M^{-1}\sum_j \hat{\epsilon}_j$ and $v = M^{-1}\sum_j(\hat{\epsilon}_j - \mu)^2$. The generation selector compares the fitted one-dimensional Gaussian $\mathcal{N}(\mu, v)$ with $\mathcal{N}(0,1)$ using three equally weighted components:

$$D_{L2} = \mu^2, \qquad D_{\mathrm{KL}} = \tfrac{1}{2}(\mu^2 + v - 1 - \log v), \qquad D_{W2} = \mu^2 + (\sqrt{v} - 1)^2. \tag{4}$$

D_{L2} measures only the patch mean, whereas D_{KL} and D_{W2} also respond to variance departures in either direction. D_{W2} is squared Gaussian Wasserstein distance. The candidate score averages these three discrepancies over patches and channels and sums them with equal weights; no weights are fitted on downstream labels. Numerical details appear in Appendix B.

These scores use only fitted means and variances. They cannot directly detect higher-order changes that preserve both moments, although such input changes may alter moments of the model's prediction. A low score denotes agreement under this proxy, not a likelihood or a preference judgment.

3.3 A Controlled Noise-Process Diagnostic

We test whether a fixed ADM ImageNet-256 model exposes controlled departures from its Gaussian forward-noise process in predicted-noise statistics. Using 128 held-out images, we keep the clean image fixed and intervene only on the injected noise, including mean shifts, within-patch variance rescaling, and moment-matched higher-order changes. At $t = 250$, predicted noise is scored over a 4×4 grid of channel-wise patches. Table 1 shows representative conditions; full interventions and controls are reported in Appendix Table S2.

Table 1: AUROC for representative controlled noise-process interventions. L2 is the squared patch mean; Energy is the full second moment. KL and W_2^2 additionally use patch variance.

Noise intervention	L2	Energy	KL	W_2^2
Mean shift $+0.3$	0.824	0.549	0.730	0.760
Residual SD $\times 0.8$	0.760	0.000	1.000	1.000
Residual SD $\times 1.2$	0.954	0.027	1.000	1.000
Moment-matched cubic tails	0.792	0.000	1.000	1.000

Across the four residual-scale interventions reported in Appendix Table S2, both KL and squared Wasserstein reach AUROC 1.000, whereas the mean-only L2 statistic is not uniformly perfect. This isolates the value of variance information without requiring any particular combination of the three score components. Moreover, moment-matched correlation and cubic-tail perturbations, although invisible to the input moments used by these formulas, become separable after passing through the

fixed predictor. Thus predicted-noise moments can expose selected higher-order departures from the forward-noise process.

These results establish statistical sensitivity under controlled interventions, not a universal OOD ordering. Broader clean-calibrated and natural-image shift results are reported in Appendix E.

3.4 GUIDANCE COMPATIBILITY

Standard CFG extrapolates the prediction as $\hat{\epsilon}_w = \hat{\epsilon}_u + w(\hat{\epsilon}_c - \hat{\epsilon}_u)$ for $w > 1$. CFG++ instead motivates manifold-constrained guidance using an interpolated clean estimate and unconditional re-noising Chung et al. (2024). This motivates testing whether guidance that better preserves the learned denoising behavior offers a more useful setting for the noise-reference score. We compare the guidance rules and separately change the scored branch under a fixed CFG sampler.

4 SELFVERIFY WITH PREDICTED-NOISE STATISTICS

The preceding diagnostic asks whether departures from training conditions appear in predicted-noise statistics. SELFVERIFY turns the same statistical object into a candidate-ranking rule. Given n trajectories generated for the same condition, it evaluates their current noise estimates at a designated sampling index, retains the lowest-scoring candidate, and continues only that sampler state. The scored quantity is always $\hat{\epsilon}$ in unit-noise coordinates; the method does not score a latent state. Generation metrics test the usefulness of the rule independently of the OOD diagnostic.

Recovering the scored noise. For epsilon-prediction models, including SDXL (Podell et al., 2023) and AudioLDM2 (Liu et al., 2024) in the evaluated configurations, the denoiser output provides $\hat{\epsilon}$ directly. For a flow model with state $x_\sigma = (1-\sigma)x_0 + \sigma\epsilon$ and velocity target $v = \epsilon - x_0$, eliminating x_0 gives

$$\hat{\epsilon} = x_\sigma + (1-\sigma)\hat{v}. \tag{5}$$

We use this conversion for SANA-base (Xie et al., 2025). More cases are given explicitly in Appendix C, together with direct scale probes.

Guidance branch. Standard CFG computes unconditional and conditional predictions and steps with

$$\hat{\epsilon}_w = \hat{\epsilon}_u + w(\hat{\epsilon}_c - \hat{\epsilon}_u). \tag{6}$$

Our standard-CFG (Ho & Salimans, 2022) selector scores this guided prediction. For the implemented CFG++ setting (Chung et al., 2024), the clean estimate uses the interpolated prediction at scale 0.6, while the step is re-noised with the unconditional estimate; we score that unconditional estimate. For SANA-base, CFG++ denotes our flow-sampler adaptation of the clean-estimate/unconditional-re-noising rule, and we do not assume that the DDIM manifold analysis transfers unchanged. A further SANA control keeps the standard CFG update fixed but scores $\hat{\epsilon}_u$, which CFG has already computed. This isolates the scored branch from the sampling update more directly than the CFG/CFG++ comparison alone.

Patch score and selection. At a designated zero-based sampling index t_p, we divide each candidate's $\hat{\epsilon}$ into channel-wise groups. For image-like grids, a grouping parameter p gives $p \times p$ values per channel; for FLUX it acts on the packed-token grid, and AudioLDM2 uses time-frequency groups. Let K be the resulting number of channel-wise groups. Within each group we estimate the mean and population variance. The candidate score averages, over groups, the equal-weight sum of three moment discrepancies:

$$S_i = \frac{1}{K}\sum_{k=1}^{K}\left[D_{L2}(\mu_{ik}, v_{ik}) + D_{\mathrm{KL}}(\mu_{ik}, v_{ik}) + D_{W2}(\mu_{ik}, v_{ik})\right].$$

The candidate minimizing S_i is kept together with its corresponding sampler state. When steps remain, only that state is continued. All three components are dimensionless moment discrepancies against the same reference, and their weights are fixed rather than fitted on downstream labels. No external preference, aesthetic, composition, or speech-quality predictor participates in selection. At $p = 1$, the within-group variance is identically zero; we therefore treat that setting as an energy-like diagnostic control rather than evidence for local distribution estimation.

Algorithm 1 Predicted-noise candidate selection at step t_p

Require: Candidate states $\{x_i\}_{i=1}^n$, current denoiser predictions, scheduler parameters, patch size p

1: **for** $i = 1, \ldots, n$ **do**
2: Obtain $\hat{\epsilon}_i$ from the designated branch and parameterization
3: Partition $\hat{\epsilon}_i$ into channel-wise patches
4: Estimate patch means and variances; compute score S_i
5: **end for**
6: $i^* \leftarrow \arg\min_i S_i$
7: Keep candidate i^* and its corresponding sampler state
8: Continue remaining steps, if any

Evaluation and computational cost. We retrospectively select from fully generated candidate pools to enable prompt-paired evaluation on identical candidates. The corresponding online n-to-one pruning budget at zero-based index t_p is

$$C_{\mathrm{SV}} = T + (n-1)(t_p + 1), \tag{7}$$

versus T for one trajectory and nT for BoN-n; two-branch CFG doubles these denoiser-evaluation counts. These are analytical compute budgets rather than measured wall-clock costs. Lower SELFVERIFY scores denote greater reference agreement, with downstream quality evaluated independently.

5 EXPERIMENTS

5.1 SETUP AND BASE-MODEL COMPARISON

We evaluate SELFVERIFY on 400 HPDv2 prompts (Wu et al., 2023) with four candidates per prompt. Our primary backbones are SANA-base 1.6B (Xie et al., 2025) and SDXL-base (Podell et al., 2023) , using 30 sampling steps with selection at $t_p = 15$. We evaluate both standard CFG and CFG++, and report two representative grouping scales, $p = 2$ and $p = 8$. The reported SELFVERIFY score uses the three-component sum $D_{L2} + D_{\mathrm{KL}} + D_{W2}$; the Hellinger component is not used in these experiments.

Generation quality is measured with HPSv2 (Wu et al., 2023), ImageReward (Xu et al., 2023), PickScore (Kirstain et al., 2023), CLIPScore (Hessel et al., 2022), and a LAION aesthetic predictor [1]. Vanilla denotes uniform selection from the same candidate pool. For comparison, gray-shaded BoN baselines use ImageReward as an external verifier, whereas SELFVERIFY uses only the diffusion model's predicted-noise statistics. Full configurations, sweeps, and uncertainty estimates are provided in the appendix B.

5.2 SELECTION UNDER STANDARD CFG

Under standard CFG, SELFVERIFY improves several downstream metrics on SANA-base (Table 2). At $p = 8$, HPSv2 increases from 0.2494 to 0.2529, while PickScore, CLIPScore, and aesthetic quality also improve. The gains are smaller at $p = 2$, but the overall pattern indicates that the predicted-noise signal can provide a useful selection criterion even without modifying the underlying sampling rule.

The effect is weaker on SDXL under the same evaluation protocol, suggesting that the usefulness of the signal depends on the interaction between the backbone and guidance configuration rather than being uniform across models.

Table 2 also compares SELFVERIFY with the gray-shaded Best-of-N baselines, which use ImageReward as an external verifier. Unlike these baselines, SELFVERIFY uses only statistics already produced by the diffusion model and can prune trajectories before generation is complete. Detailed metric-wise uncertainty estimates are provided in Appendix F.

[1] https://github.com/christophschuhmann/improved-aesthetic-predictor.

Table 2: Predicted-noise selection on 400 HPDv2 prompts, $n = 4$, $T = 30$, and $t_p = 15$. Each row is evaluated on the same candidate pool within its backbone/guidance condition. Gray-shaded rows use an external verifier for selection; IR denotes ImageReward and † marks evaluation on the same metric used for selection. SELFVERIFY uses no external verifier. Work counts conditional/unconditional denoiser evaluations separately.

Method	NFEs	HPSv2 ↑	IR ↑	PickScore ↑	CLIP ↑	Aesthetic ↑
SANA / CFG 4.5						
Vanilla	60	0.2494	0.0859	20.6811	26.2470	5.8021
BoN-2 (IR)	120	0.2596	0.4306†	20.9741	26.8448	5.9076
BoN-4 (IR)	240	0.2662	0.6964†	21.1789	27.2298	5.9758
SELFVERIFY $p = 2$	156	0.2513	0.0976	20.8115	26.5575	5.8576
SELFVERIFY $p = 8$	156	0.2529	0.1245	20.8292	26.4854	5.8578
SANA / CFG++ 0.6						
Vanilla	60	0.2246	-0.4509	19.8188	24.7225	5.5360
BoN-2 (IR)	120	0.2398	-0.0342†	20.2553	25.7462	5.6985
BoN-4 (IR)	240	0.2512	0.3123†	20.5749	26.4690	5.7984
SELFVERIFY $p = 2$	156	0.2346	-0.2377	20.1825	25.6199	5.6945
SELFVERIFY $p = 8$	156	0.2378	-0.1997	20.2131	25.4603	5.7133
SDXL / CFG 7.5						
Vanilla	60	0.2872	0.9584	22.6129	30.3498	6.0322
BoN-2 (IR)	120	0.2924	1.1753†	22.7325	30.5098	6.0521
BoN-4 (IR)	240	0.2955	1.3224†	22.8059	30.7006	6.0610
SELFVERIFY $p = 2$	156	0.2879	0.9646	22.6174	30.2388	6.0340
SELFVERIFY $p = 8$	156	0.2873	0.9663	22.6265	30.4176	6.0384
SDXL / CFG++ 0.6						
Vanilla	60	0.2742	0.7842	22.4229	29.8406	6.0664
BoN-2 (IR)	120	0.2809	1.0461†	22.5844	30.0922	6.0868
BoN-4 (IR)	240	0.2856	1.2255†	22.7119	30.3524	6.0953
SELFVERIFY $p = 2$	156	0.2768	0.8110	22.4546	29.8202	6.0766
SELFVERIFY $p = 8$	156	0.2753	0.7612	22.4602	29.8195	6.0729

Table 3: Guidance and scored-branch comparison at $t_p = 15$. HPSv2 gains are measured relative to each setting's own vanilla pool. CFG scales are 4.5 for SANA and 7.5 for SDXL; CFG++ uses 0.6. Intervals are paired within each setting, not confidence intervals on differences between settings.

Backbone / guidance / score	Δ HPSv2, $p = 2$ [CI]	Δ HPSv2, $p = 8$ [CI]
SANA / CFG, guided score	$+0.0019$ $[-0.0004, +0.0042]$	$+0.0035$ $[+0.0013, +0.0057]$
SANA / CFG, uncond. score	$+0.0062$ $[+0.0039, +0.0085]$	$+0.0072$ $[+0.0050, +0.0095]$
SANA / CFG++	$+0.0100$ $[+0.0072, +0.0128]$	$+0.0132$ $[+0.0102, +0.0161]$
SDXL / CFG	$+0.0007$ $[-0.0009, +0.0022]$	$+0.0001$ $[-0.0014, +0.0017]$
SDXL / CFG++	$+0.0025$ $[+0.0008, +0.0043]$	$+0.0010$ $[-0.0007, +0.0028]$

5.3 GUIDANCE COMPATIBILITY: CFG++ AND THE SCORED BRANCH

The selection signal becomes substantially stronger for SANA-base with our CFG++ implementation (Tables 2 and 3). At $p = 8$, the HPSv2 gain increases from $+0.0035$ under standard CFG to $+0.0132$ under CFG++, and all five downstream evaluation metrics improve relative to the corresponding vanilla pool. A similar pattern is observed at $p = 2$. This suggests that the usefulness of the predicted-noise typicality signal is sensitive to the guidance dynamics under which the trajectory is generated.

To separate the effect of the sampling update from the prediction being scored, we additionally keep standard CFG sampling fixed and score the unconditional prediction, which is already available from the CFG computation. This increases the HPSv2 gain to $+0.0072$ at $p = 8$, compared with $+0.0035$ when scoring the guided prediction (Table 3). Thus, part of the improvement associated with CFG++ can be attributed to the choice of scored branch rather than to the sampling rule alone.

Both the guidance rule and the scored branch affect selection; this does not imply that the score measures manifold distance. Detailed intervals are in Appendix F.

Table 4: Compositional generation on GenEval2 with 800 prompts, $n = 4$, $T = 30$, and $t_p = 15$. Changes are relative to the same-pool candidate mean; brackets denote pointwise 95% prompt-bootstrap intervals Soft-TIFA is scaled by 100, and BoN-4 selects by ImageReward.

Selector	Δ Soft-TIFA [95% CI]	Δ HPSv2 [95% CI]
SANA-base, CFG 4.5 (vanilla Soft-TIFA: 48.45)		
SELFVERIFY $p = 2$	$+1.47$ [$+0.52$, $+2.40$]	$+0.0061$ [$+0.0042$, $+0.0080$]
SELFVERIFY $p = 8$	$+1.47$ [$+0.55$, $+2.37$]	$+0.0080$ [$+0.0060$, $+0.0099$]
BoN-4 (IR)	$+7.68$ [$+6.82$, $+8.52$]	$+0.0258$ [$+0.0243$, $+0.0273$]
SANA-base, CFG++ 0.6 (vanilla Soft-TIFA: 40.73)		
SELFVERIFY $p = 2$	$+4.96$ [$+4.05$, $+5.87$]	$+0.0190$ [$+0.0170$, $+0.0210$]
SELFVERIFY $p = 8$	$+4.71$ [$+3.77$, $+5.67$]	$+0.0172$ [$+0.0151$, $+0.0193$]
BoN-4 (IR)	$+9.59$ [$+8.74$, $+10.44$]	$+0.0298$ [$+0.0283$, $+0.0315$]
SDXL-base, CFG 7.5 (vanilla Soft-TIFA: 50.66)		
SELFVERIFY $p = 2$	$+0.07$ [-0.74, $+0.90$]	$+0.0007$ [-0.0007, $+0.0020$]
SELFVERIFY $p = 8$	-0.39 [-1.21, $+0.43$]	$+0.0023$ [$+0.0010$, $+0.0035$]
BoN-4 (IR)	$+5.17$ [$+4.42$, $+5.99$]	$+0.0127$ [$+0.0116$, $+0.0139$]
SDXL-base, CFG++ 0.6 (vanilla Soft-TIFA: 49.03)		
SELFVERIFY $p = 2$	-0.86 [-1.68, -0.03]	-0.0003 [-0.0017, $+0.0012$]
SELFVERIFY $p = 8$	$+0.91$ [$+0.08$, $+1.72$]	$+0.0043$ [$+0.0030$, $+0.0056$]
BoN-4 (IR)	$+5.37$ [$+4.61$, $+6.13$]	$+0.0146$ [$+0.0135$, $+0.0158$]

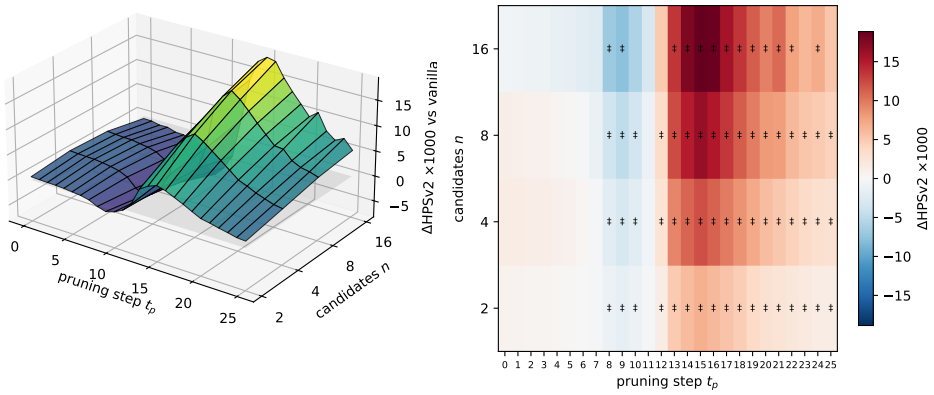

Figure 2: SANA-base predicted-noise selection with CFG++: HPSv2 gain over the subset-mean vanilla baseline. Left: candidate-count surface; right: heat map, with Bonferroni-significant cells marked over 104 displayed cells per metric. Values are $\Delta\text{HPSv2} \times 1000$. The window is $t_p = 0$–25; remaining steps are reported in the Appendix. Faces connect measured grid values.

5.4 COMPOSITIONAL GENERATION

We evaluate compositional alignment on 800 GenEval2 prompts Kamath et al. (2025); Table 4 summarizes changes in Soft-TIFA and HPSv2 relative to the same-pool candidate mean. SANA-base shows the clearest compositional gains, especially under CFG++: SELFVERIFY improves Soft-TIFA by 4.96 and 4.71 points at $p = 2$ and $p = 8$, with corresponding HPSv2 gains of 0.0190 and 0.0172; both Soft-TIFA effects survive the sweep-level Bonferroni correction. Standard CFG gives smaller positive gains. SDXL is weaker and less consistent, although CFG++ at $p = 8$ improves both Soft-TIFA ($+0.91$) and HPSv2 ($+0.0043$). Appendix F.3 reports the full sweep and the $p = 1$ control.

5.5 CANDIDATE COUNT AND SELECTION TIME

We next examine inference-time scaling on a separate SANA-base run with CFG++, using candidate pools of size $n \in \{2, 4, 8, 16\}$ and scoring at multiple intermediate sampling steps (Figure 2). At $t_p = 15$, the HPSv2 gain increases from 0.0066 for two candidates to 0.0122, 0.0164, and 0.0189 for four, eight, and sixteen candidates, respectively.

This trend shows that SELFVERIFY can make increasingly effective use of a larger candidate pool: as more trajectories are available, the typicality score has more opportunity to identify a favorable candidate. The effect should be interpreted as improved exploitation of a fixed candidate pool rather than as an unbounded scaling law.

Selection time is also important. As shown in Figure 2, the strongest gains are concentrated around intermediate denoising steps rather than appearing uniformly throughout the trajectory. This is consistent with the idea that the predicted-noise statistics become informative only after sufficient structure has emerged, while very late selection leaves less opportunity for pruning to reduce computation.

Complete step/count surfaces for ImageReward, PickScore, CLIPScore, and aesthetic quality, together with the late-step results, are reported in Appendix F.5.

5.6 SPEECH EXTENSION

We further test whether the same predicted-noise selection rule transfers beyond image generation by applying SELFVERIFY to AudioLDM2 (Liu et al., 2024) (Table 5). We use four candidates and evaluate speech naturalness with UTMOS (Saeki et al., 2022), which is never observed by SELFVERIFY during selection.

Table 5: AudioLDM2 predicted-noise selection. UTMOS is an evaluation metric, not the SELFVERIFY selection score. Differences are against the same-pool vanilla mean; the oracle directly selects on UTMOS.

Method	UTMOS ↑	Δ UTMOS
Vanilla	3.557	—
SELFVERIFY $M = 4$	3.585	+0.028
SELFVERIFY $M = 64$	3.575	+0.018
UTMOS (BoN-4)	3.772	+0.215

At the reference setting, UTMOS increases from 3.557 for vanilla selection to 3.585 with SELFVERIFY. A second grouping scale also yields an improvement, indicating that the effect is not specific to a single grouping choice. These results provide evidence that predicted-noise statistics can act as a useful internal selection signal for a modality and downstream quality metric different from those used in the main image experiments.

The gray-shaded oracle in Table 5 directly selects candidates using UTMOS itself and therefore represents an external-verifier upper reference within the same candidate pool. SELFVERIFY, in contrast, does not query a speech-quality model during selection. Full grouping and selection-time sweeps are provided in Appendix G.2.

LIMITATIONS

SELFVERIFY is an operational typicality proxy rather than a likelihood, formal typical-set test, manifold-distance estimator, or general OOD detector. Evidence is strongest on SANA-base; SDXL and few-step distilled transfer are less consistent, and audio gains involve a quality–alignment trade-off. The stronger relative gains under CFG++ do not isolate a manifold effect because guidance strength, sampling rule, scored branch, and baseline quality also change. Controlled interventions show that variance-sensitive terms capture information absent from the patch-mean statistic, but generation ablations do not establish that the equal-weight combination of the three moment discrepancies is necessary or optimal. Evaluation is retrospective on fixed candidate pools, compute estimates are analytical, and several sweep and reporting choices are exploratory.

CONCLUSION

SELFVERIFY uses predicted-noise statistics to select diffusion trajectories without an external reward model. Its main empirical support is on base models under guidance: SANA-base benefits under standard CFG and shows stronger relative gains with our CFG++ implementation, while a scored-branch control clarifies the role of the prediction used for selection. Candidate-count scaling and speech-naturalness results extend this evidence, whereas the few-step distilled evaluation marks a current limitation. The resulting picture is a guidance-dependent typicality proxy whose selection can improve downstream quality, with the manifold interpretation serving as a motivated explanation rather than a measured property.

AI USE STATEMENT

We used generative AI tools for language editing, literature search, and implementation assistance. All AI-assisted outputs were reviewed and verified by the authors, and all final scientific judgments and evaluations were made by the authors.

REPRODUCIBILITY STATEMENT

We provide the implementation details, experimental settings, evaluation protocols, and additional results needed to reproduce our experiments in the main text and appendix. In particular, Appendix B describes the experimental configurations and statistical procedures, while Appendix C details the model-specific predicted-noise extraction and implementation.

REFERENCES

Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Abdelhamed_A_High-Quality_Denoising_CVPR_2018_paper.html.

Sumukh K Aithal, Pratyush Maini, Zachary Chase Lipton, and J Zico Kolter. Understanding hallucinations in diffusion models through mode interpolation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=aNTnHBkw4T.

Cosmin I Bercea, Michael Neumayr, Daniel Rueckert, and Julia A Schnabel. Mask, stitch, and re-sample: Enhancing robustness and generalizability in anomaly detection through automatic diffusion models. *arXiv preprint arXiv:2305.19643*, 2023.

Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. Sana-sprint: One-step diffusion with continuous-time consistency distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16185–16195, 2025.

Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. Cfg++: Manifold-constrained classifier free guidance for diffusion models. *arXiv preprint arXiv:2406.08070*, 2024.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Guilherme Fernandes, Vasco Ramos, Regev Cohen, Idan Szpektor, and João Magalhães. Latent beam diffusion models for generating visual sequences, 2025. URL https://arxiv.org/abs/2503.20429.

Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008. doi: 10.1109/TIP.2008.2001399.

Alvin Heng, Alexandre H. Thiery, and Harold Soh. Out-of-distribution detection with a single unconditional diffusion model. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 43952–43974. Curran Associates, Inc.,

2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/4dc37a7bc61057252ce043fa3b83aac2-Paper-Conference.pdf.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022. URL https://arxiv.org/abs/2104.08718.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. Geneval 2: Addressing benchmark drift in text-to-image evaluation, 2025. URL https://arxiv.org/abs/2512.16853.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=G5RwHpBUv0.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis, 2021. URL https://arxiv.org/abs/2009.09761.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2871–2883, 2024.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl, 2025. URL https://arxiv.org/abs/2505.05470.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Victor Livernoche, Vineet Jain, Yashar Hezaveh, and Siamak Ravanbakhsh. On diffusion modeling for anomaly detection. *arXiv preprint arXiv:2305.18593*, 2023.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 50611–50649, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7e9c2053258b1bdd32ff2654802cd594-Paper-Conference.pdf.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Scaling inference time compute for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2523–2534, June 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. In *Proceedings of Interspeech*, pp. 4521–4525, 2022. doi: 10.21437/Interspeech.2022-439.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Kaixuan Wei, Ying Fu, Jiaolong Yang, and Hua Huang. A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2758–2767, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Wei_A_Physics-Based_Noise_Formation_Model_for_Extreme_Low-Light_Raw_Denoising_CVPR_2020_paper.html.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Enze Xie, Junsong Chen, Yuyang Zhao, Jincheng YU, Ligeng Zhu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, Bingchen Liu, Daquan Zhou, and Song Han. SANA 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=27hOkXzy9e.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 15903–15935, 2023.

Hang Yao, Ming Liu, Zhicun Yin, Zifei Yan, Xiaopeng Hong, and Wangmeng Zuo. Glad: Towards better reconstruction with global and local adaptive diffusion models for unsupervised anomaly detection. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2024.

Hui Zhang, Zheng Wang, Dan Zeng, Zuxuan Wu, and Yu-Gang Jiang. Diffusionad: Norm-guided one-step denoising diffusion for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Jiachao Zhang, Keigo Hirakawa, and Xiaodan Jin. Quantile analysis of image sensor noise distribution. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1598–1602, 2015. doi: 10.1109/ICASSP.2015.7178240.

A Appendix

The appendix provides experimental details (Appendix B), noise extraction and validation (Appendix C), statistical derivations (Appendix D), distribution diagnostics (Appendix E), additional base-model results (Appendix F), and transfer evaluations (Appendix G). All generation selectors use predicted noise. The revised generation score is the equal-weight sum of squared patch mean, Gaussian KL, and squared Wasserstein; the Hellinger-like component is omitted.

B Experimental Details

B.1 Backbones and Sampling Configurations

The primary image experiments use 400 HPDv2 prompts, base seed 42, four candidates, 30 sampling steps, and selection index $t_p = 15$. SDXL-base uses DDIM with CFG 7.5 or CFG++ 0.6; SANA-base 1.6B uses the implemented flow sampler with CFG 4.5 or its CFG++ adaptation at 0.6. The SANA unconditional-scoring control preserves the standard CFG sampling rule. Logged selection indices are 5, 10, 15, 20, and 25; patch sizes are 1, 2, 4, 8, 16, and 32. Within each channel, an image patch contains p^2 values. FLUX grouping uses its packed-token grid, so p is not a universal physical-image scale across backbones.

The SANA-base surface uses a separate 400-prompt run with 16 candidates, CFG++ 0.6, and all 30 sampling indices recorded. It fixes $p = 8$. The four-candidate subset estimate from this run is not the same sample as the independent four-candidate main run. The supplementary distilled HPDv2 boundary evaluation uses four candidates and $(T, t_p) = (1, 0)$ for SDXL-Turbo, $(2, 1)$ for SANA-Sprint, $(8, 4)$ for Z-Image-Turbo, and $(4, 2)$ for FLUX-Schnell.

The compositional study uses 800 GenEval2 prompts per configuration, four candidates, seed 42, and the same base-model samplers and guidance settings. All 800 prompts have complete scores for all four candidates. Prompt indices, question counts, and candidate image paths were checked against the benchmark and evaluation mappings. Soft-TIFA is the mean of question-level scores within each prompt, scaled by 100. Its confidence intervals use 5,000 prompt-bootstrap resamples. Sweep significance uses two-sided paired t-tests with Bonferroni correction over the 30 step/patch cells separately for each configuration and metric. The reported intervals are pointwise, not multiplicity-adjusted.

B.2 Evaluation and Statistical Units

HPSv2, ImageReward, PickScore, CLIPScore, and the saved aesthetic-predictor labels are joined to candidates by identity. Vanilla is the mean quality of all candidates, estimating uniform selection. BoN-k selects by ImageReward and averages over all k-subsets of the same four-candidate pool. Evaluation on ImageReward is marked separately from transfer to the other metrics.

Pointwise 95% intervals use 5,000 bootstrap resamples of prompts. A paired difference is computed against the same prompt's candidate-mean baseline before resampling. These intervals condition on the generated pools; they do not include training uncertainty or independent image-generation seeds. Sweep cells and the distilled boundary evaluation are exploratory, not a held-out hyperparameter-selection evaluation. Surface cells average 256 random candidate subsets for $n < 16$ and use the full pool for $n = 16$; subset identities are shared across steps. Heat-map markers use a Bonferroni correction over the 104 displayed step/count cells per metric, not over model or metric selection.

B.3 Speech Configuration

AudioLDM2 uses `anhnct/audioldm2_gigaspeech`, guidance 1.0, four candidates, 50 steps, and reference selection index 25. The reporting set contains 399 prompts common to seeds 43–45; seed 42 is excluded from the performance estimate. Seed outcomes are averaged within each prompt before bootstrap resampling. UTMOS uses `utmos22_strong` from `tarepan/SpeechMOS:v1.2.0` on 16-kHz audio Saeki et al. (2022). It is used for evaluation, except in the explicitly labeled oracle. CLAP is a secondary description-alignment measure using normalized embeddings from `laion/clap-htsat-unfused`, with audio resampled to 48 kHz.

Group sizes $M = 1, 4, 16, 64, 416, 1248$ correspond to the logged single-element, 2×2, 2×8, 4×16, 26×16, and 78×16 groupings. The speech grid uses indices 5, 12, 25, 37, and 45. The extraction probes in Appendix C are separate from the performance reporting set.

B.4 Score Implementation and Computational Cost

The three generation components—squared patch mean, Gaussian KL, and squared Wasserstein—are averaged over patches and channels, then summed with equal weights. The implementation regularizes variance with 10^{-6} for logarithms and square roots. The revised selector does not include the Hellinger-like term. At $M = 1$, the within-group variance is zero, making the score an energy-like control rather than a reliable local variance estimate.

All reported generation results select retrospectively from fully generated pools. Counting the denoiser call at the zero-based selection index, online n-to-one pruning would use $T + (n-1)(t_p + 1)$ candidate-steps. Two-branch CFG doubles the model-example evaluation count. Thus the main four-candidate runs correspond to 156 evaluations, versus 60 for vanilla, 120 for BoN-2, and 240 for BoN-4. Audio's corresponding counts are 128, 50, and 200 for SelfVerify, vanilla, and BoN-4. These counts exclude decoding, external-verifier cost, and batching effects; they are not measured runtime speedups. At the final step, pruning saves no denoiser work.

C Noise Extraction and Validation

SelfVerify scores the model's noise estimate $\hat{\epsilon}$ at the pruning step t_p (Section 4). Recovering $\hat{\epsilon}$ depends on the pipeline: each backbone predicts a different quantity, several pipelines rescale or re-sign the network output before the scheduler consumes it, and the callbacks used in the evaluated pipelines do not supply the raw denoiser output. This appendix states, for every backbone, the parameterization, the exact transformation from the network output to $\hat{\epsilon}$, the guidance branch that is scored, and moment-based scale statistics of the converted estimate at the scored step. Those statistics are what the three score components consume; they check the scale of the conversion and make no claim that the estimate is Gaussian. No backbone requires an extra network evaluation: $\hat{\epsilon}$ is assembled from tensors the sampler computes anyway.

C.1 Parameterizations

Let x denote the noisy state handed to the denoiser and o the raw network output.

- **ϵ-prediction** (DDPM/DDIM, $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon$): the network output is the estimate, $\hat{\epsilon} = o$.
- **Flow matching** ($x_\sigma = (1 - \sigma)x_0 + \sigma\epsilon$, velocity target $v = \epsilon - x_0$): the Euler update is $x_{\sigma'} = x_\sigma + (\sigma' - \sigma)\hat{v}$, and
$$\hat{\epsilon} = x_\sigma + (1 - \sigma)\, \hat{v}. \tag{S1}$$
- **TrigFlow / SCM** ($x_s = \cos s\, x_0 + \sin s\, z$, $z \sim \mathcal{N}(0, \sigma_d^2 I)$): the scheduler forms $\hat{x}_0 = \cos s\, L - \sin s\, M$ from its state L and model output M, so the noise estimate in unit-noise coordinates is
$$\hat{\epsilon} = \big(\sin s\, L + \cos s\, M\big)/\sigma_d = \big(L - \cos s\, \hat{x}_0\big)/(\sigma_d \sin s). \tag{S2}$$

In every case $\hat{\epsilon}$ is the residual $(x - \text{signal coefficient} \cdot \hat{x}_0)/\text{noise coefficient}$, i.e. the same object as in Eq. equation S1–equation S2 for the respective coordinates.

C.2 Guidance branch

With classifier-free guidance the network is evaluated on an unconditional and a conditional branch. Under plain CFG the sampler steps with the guided estimate $\hat{\epsilon}_u + w(\hat{\epsilon}_c - \hat{\epsilon}_u)$; we report both scoring that guided estimate and scoring the unconditional estimate $\hat{\epsilon}_u$, which CFG computes anyway. Under CFG++ (Chung et al., 2024) the step is re-noised with $\hat{\epsilon}_u$, and that is the estimate we score. The reference distribution is always $\mathcal{N}(0, I)$ on the estimate; the sampling update itself is never modified.

C.3 PER-BACKBONE TRANSFORMATIONS

Table S1 lists, for each backbone, the sampler, the number of steps T and the scored step t_p, and the transformation applied to the raw network output. Three pipeline details are easy to miss and change the result:

1. **SDXL-Turbo** (`EulerAncestralDiscrete`, ϵ-prediction) feeds the network $x_t/\sqrt{\sigma_t^2+1}$; the output is $\hat{\epsilon}$ unchanged.
2. **Z-Image-Turbo** predicts $x_0 - \epsilon$: the pipeline negates the network output before the flow-matching step, so $\hat{v} = -o$ in Eq. equation S1. FLUX-Schnell passes o to the scheduler unchanged. Because the sign is invisible at $\sigma = 1$, our implementation verifies at run time that the post-step latent equals $x_\sigma + (\sigma' - \sigma)\hat{v}$ (relative error $\approx 2 \times 10^{-3}$ in bf16; the run aborts above 10^{-2}).
3. **SANA-Sprint** (`SCM`, $\sigma_d = 0.5$) keeps its scheduler state L at scale σ_d, feeds the network $x_{\text{in}} = (L/\sigma_d)\, c$ with $c = \sqrt{\tau^2 + (1-\tau)^2}$, $\tau = \sin s/(\cos s + \sin s)$, and passes $M = \sigma_d\big[(1-2\tau)x_{\text{in}} + (1-2\tau+2\tau^2)\, o\big]/c$ to the scheduler. Substituting into Eq. equation S2 gives $\hat{\epsilon} = \sin s\, x_{\text{in}}/c + \cos s\big[(1-2\tau)x_{\text{in}} + (1-2\tau+2\tau^2)o\big]/c$. At the first step ($\tau = 1$) this reduces to x_{in}, so a first-step check does not exercise the transformation; the second step does.

C.4 SCALE STATISTICS OF THE CONVERTED ESTIMATE

Columns 6–9 of Table S1 characterize the scale of $\hat{\epsilon}$ at t_p over the same patches the score uses at its largest scale ($p \times p$ patches within one channel, $M = p^2$; $p = 32$, $M = 1024$ for the image backbones; 78×16 time–frequency tiles, $M = 1248$, for AudioLDM2), from two sources. The logged runs record only patch-averaged divergences, from which exactly two patch statistics follow: the root-mean-square patch mean, $\mathrm{RMS}(\hat{\mu}) = \sqrt{\overline{\ell_2}}$, and the root-mean-square deviation of the patch standard deviation from one, $\mathrm{RMS}(\hat{\sigma} - 1) = \sqrt{\overline{W_2 - \ell_2}}$, pooled over patches, candidates and prompts (columns 6–7). These are not the mean absolute patch mean or the mean patch standard deviation, since mean, square and square root do not commute; in particular $\mathrm{RMS}(\hat{\sigma} - 1)$ contains the patch-to-patch dispersion of $\hat{\sigma}$, which for an exactly $\mathcal{N}(0, I)$ field is already $\approx 1/\sqrt{2M}$ (0.022 at $M = 1024$), so it only upper-bounds $|\overline{\hat{\sigma}} - 1|$. We therefore also re-generated eight prompts per backbone with the same pipeline and measured the per-patch statistics of the $\hat{\epsilon}$ tensor directly (columns 8–9). The direct RMS values reproduce the logged ones to within 0.007 and typically 0.002, providing a consistency check; the direct means characterize the smaller probe sample. FLUX-Schnell ($\overline{\hat{\sigma}} = 1.003$) and AudioLDM2 without guidance (0.992) have mean patch standard deviations close to one in this probe; SDXL-Turbo and Z-Image-Turbo sit at 0.97, SDXL-base at 0.94, and SANA-Sprint at its second step at 0.89; SANA-base under either guidance rule is *above* one (1.01–1.03), as is AudioLDM2 under CFG 3.5 (1.05, together with elevated absolute patch means, $|\overline{\hat{\mu}}| = 0.070$ against a reference of 0.023). A departure from unit scale alone does not establish a conversion error. Under ideal squared-error training the optimal predictor is a conditional mean, whose output variance can be reduced; finite-model errors, guidance, and distillation can also alter the measured statistics. Nor is the departure common to all candidates: the last column gives the standard deviation across candidates of the per-candidate mean patch standard deviation, which is negligible for SDXL-Turbo and FLUX (0.003–0.004) but reaches 0.020 for SANA-Sprint and 0.03–0.04 for SANA-base under guidance. Candidate-to-candidate differences of this size enter the KL and W_2 terms and are part of what the score ranks; the reference $\mathcal{N}(0, I)$ is a fixed anchor against which the candidates of a prompt are compared under identical conditions, and we make no assumption that their scales coincide.

C.5 IMPLEMENTATION AND ADDITIONAL PARAMETERIZATIONS

The distilled-backbone wrapper captures the current network input and output without changing sampling. The remaining evaluated backends construct $\hat{\epsilon}$ inline. For flow pipelines, a runtime reconstruction check compares the predicted Euler update against the scheduler update; this verifies the sign convention beyond the initial step. The SANA-Sprint conversion follows its input/output affine

Table S1: How $\hat{\epsilon}$ is obtained from each backbone, and the scale of the converted estimate at the scored step. Columns 6–7: root-mean-square patch mean and root-mean-square deviation of the patch standard deviation from one, recovered exactly from the logged, patch-averaged ℓ_2 and W_2 components of the full runs (400 prompts; AudioLDM2 399 prompts per seed). Columns 8–9: direct per-patch measurements from re-generating 8 prompts $\times$ 4 candidates per backbone with the same pipeline: mean absolute patch mean and mean patch standard deviation, and the standard deviation across candidates of the per-candidate mean patch standard deviation. Patches are 32×32 within one channel ($M = 1024$) for the image backbones and 78×16 tiles ($M = 1248$) for AudioLDM2; the reference rows give the same statistics for an exactly standard-normal field.

Backbone	Sampler / parameterization	T	t_p	$\hat{\epsilon}$ from network output o	RMS($\hat{\mu}$)	RMS($\hat{\sigma}-1$)	$\overline{\lvert\hat{\mu}\rvert}$ / $\overline{\hat{\sigma}}$ (direct)	sd$_{\text{cand}}(\overline{\hat{\sigma}})$
Reference: $\mathcal{N}(0, I)$ field, $M = 1024$ (image rows)					0.031	0.022	0.025 / 0.999	–
SDXL-Turbo	EulerAncestral, ϵ-pred.	1	0	o	0.035	0.036	0.028 / 0.969	0.003
SANA-Sprint	SCM (TrigFlow), σ_d=0.5	2	1	Eq. equation S2 with the pipeline affine (item 3)	0.022	0.114	0.018 / 0.889	0.020
Z-Image-Turbo	FlowMatchEuler, shift 3	8	4	$x_\sigma + (1-\sigma)(-o)$	0.034	0.043	0.027 / 0.966	0.006
FLUX-Schnell	FlowMatchEuler, shift 1	4	2	$x_\sigma + (1-\sigma)\, o$	0.043	0.024	0.029 / 1.003	0.004
SANA-base 1.6B, CFG++ 0.6	FlowMatchEuler, shift 3	30	15	$x_\sigma + (1-\sigma)\, o_u$ (unconditional)	0.037	0.045	0.028 / 1.019	0.032
SANA-base 1.6B, CFG 4.5	FlowMatchEuler, shift 3	30	15	$x_\sigma + (1-\sigma)[o_u + w(o_c - o_u)]$	0.035	0.059	0.027 / 1.031	0.039
SANA-base 1.6B, CFG 4.5	FlowMatchEuler, shift 3	30	15	$x_\sigma + (1-\sigma)o_u$	0.036	0.048	0.028 / 1.013	0.040
SDXL-base 1.0, CFG++ 0.6	DDIM, ϵ-pred.	30	15	o_u (unconditional)	0.024	0.063	0.020 / 0.940	0.013
SDXL-base 1.0, CFG 7.5	DDIM, ϵ-pred.	30	15	guided	0.026	0.060	0.020 / 0.946	0.014
Reference: $\mathcal{N}(0, I)$ field, $M = 1248$ (AudioLDM2 rows)					0.028	0.020	0.023 / 0.999	–
AudioLDM2-TTS, CFG-free	DDIM, ϵ-pred.	50	25	o	0.038	0.024	0.029 / 0.992	0.014
AudioLDM2-TTS, CFG 3.5	DDIM, ϵ-pred.	50	25	guided	0.096	0.071	0.070 / 1.053	0.018

transformations and is examined through the direct statistical probes; it does not use the same Euler reconstruction check.

For a clean-data predictor f_θ in coordinates $x_t = \alpha_t x_0 + \sigma_t \epsilon$, the corresponding residual is $(x_t - \alpha_t f_\theta)/\sigma_t$ for $\sigma_t > 0$. This identity also applies to compatible consistency or distilled clean-data parameterizations Song et al. (2023); Sauer et al. (2024), but their training objectives need not make the result a posterior-mean noise estimate. For an optimal squared-error flow predictor, the conversion in Eq. equation S1 equals $\mathbb{E}[\epsilon \mid x_t]$. Neither identity establishes Gaussian outputs or an OOD ordering; Appendix D explains that distinction.

D Statistical Motivation and Scope

The main text retains the mixture model and conceptual figure; this appendix develops the training correspondence and the limits of the resulting interpretation. In the figure, a' denotes the noise draw entering the mixture. Here a_{ref} denotes an independent reference used only for the output-moment identity.

D.1 Conditional Reconstruction and Output Moments

We first describe the learned conditional estimator and the statistics available from its outputs. The variables below represent the training process, rather than prescribing an image-replacement experiment.

Generative Process and Learning Objective. Let $a \in \mathbb{R}^d$ be a latent variable representing a signal, drawn from a standard multivariate Gaussian distribution $a \sim \mathcal{A} = \mathcal{N}(0, I_d)$. Let $b \in \mathbb{R}^d$ be a structured data component, drawn from an arbitrary high-dimensional data distribution B with mean μ_b and covariance Σ_b. We assume a and b are statistically independent.

The model input $x \in \mathbb{R}^d$ is generated by a linear combination of these components, $x = f(a, b) = c_1 a + c_2 b$, where $0 < c_1, c_2 \leq 1$ are scalar coefficients controlling the influence of a and b. We consider a model $M : \mathbb{R}^d \to \mathbb{R}^d$ that learns to recover a from the mixture x. The optimal model M^* minimizes the conditional reconstruction MSE:

$$M^* = \operatorname*{argmin}_M \mathcal{L}(M), \quad \mathcal{L}(M) = \mathbb{E}_{a,b}\Big[\|a - M(c_1 a + c_2 b)\|^2\Big] \tag{S3}$$

Under standard assumptions, this yields the posterior mean $M^*(x) = \mathbb{E}[a \mid x]$. As a diagnostic of output second moments, we evaluate against an independent reference $a_{\text{ref}} \sim \mathcal{A}$, drawn indepen-

dently of (a, b), $R(M)$:

$$\mathbb{E}_{a,b,a_{\text{ref}}}\Big[\|a_{\text{ref}} - M(c_1 a + c_2 b)\|^2\Big] = \underbrace{\mathbb{E}\big[\|a_{\text{ref}}\|^2\big]}_{=d,\ \text{constant}} + \mathbb{E}_{a,b}\Big[\|M(c_1 a + c_2 b)\|^2\Big] \tag{S4}$$

where the decomposition follows from the independence of a_{ref} and (a, b). Under this metric, comparing any two input conditions reduces to comparing the second moment of the model outputs.

From training conditions to a statistical-shift hypothesis. Let P_X denote the input distribution induced by the training mixture and Q_X the input distribution under a different condition. Holding the trained predictor fixed, define the output-moment statistic

$$R(P_X) = \mathbb{E}_{x\sim P_X}\|M^*(x)\|^2, \qquad R(Q_X) = \mathbb{E}_{x\sim Q_X}\|M^*(x)\|^2.$$

An input shift may arise from a change in the data distribution, the noise process, or their relationship. Our hypothesis is that such a departure can appear in the distribution of the predicted noise, including its local means and variances. The model is still applying the conditional estimator learned under the original training conditions; it has not been refitted to the new condition.

This motivates inspecting output statistics, but does not determine the sign of $R(Q_X) - R(P_X)$ or of a Gaussian-reference divergence change. The independent-reference loss above is minimized by a zero output, so it is not itself a distribution-matching objective. Section D.4 illustrates the absence of a universal ordering using a fixed data-component replacement as a special case. The controlled experiment instead perturbs the noise process to isolate particular statistical departures while holding clean images fixed.

D.2 Connection to Diffusion Noise Prediction

Let the objective from our assumed model, M, be defined as:

$$\mathcal{L}(M) = \mathbb{E}_{t\sim p(t)}\Big[\mathbb{E}_{a\sim\mathcal{N}(0,\mathbf{I}),b\sim q}\Big[\|a - M(c_{1,t}a + c_{2,t}b, c_{1,t}, c_{2,t})\|^2\Big]\Big] \tag{S5}$$

where $(c_{1,t}, c_{2,t})$ are coefficients indexed by a timestep t drawn from a distribution $p(t)$.

We begin by establishing a set of one-to-one correspondences to align our proposed circumstances with the DDPM framework. First, we map the variable $a \leftrightarrow \epsilon$ and $b \leftrightarrow x_0$, where both a and ϵ are sampled from $\mathcal{N}(0, \mathbf{I})$, and both b and x_0 are from the data distribution q. Second, the coefficients $(c_{1,t}, c_{2,t})$ are defined to match the DDPM schedule: $c_{1,t} := \sqrt{1-\bar{\alpha}_t}$ and $c_{2,t} := \sqrt{\bar{\alpha}_t}$. Given that the timestep t uniquely determines these coefficients, our model M (conditioned on coefficients) becomes functionally identical to the DDPM model ϵ_θ (conditioned on t). This equivalence is formally expressed as $M(\cdot, c_{1,t}, c_{2,t}) \equiv \epsilon_\theta(\cdot, t)$.

By substituting these mappings directly into our redefined objective, we obtain:

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim p(t)}\Big[\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I}),x_0\sim q}\Big[||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2\Big]\Big] \tag{S6}$$

Substitution recovers the DDPM noise-prediction objective; the mixture formulation is a reparameterization of that training setup. The correspondence makes the training dependence explicit: under squared-error optimality, $\epsilon_\theta(x_t, t)$ estimates $\mathbb{E}_{\text{train}}[\epsilon \mid x_t, t]$. If the input process changes, the same predictor is evaluated under a different input law, and its output statistics may change. This is the basis for testing noise predictions as a source of distributional-conformity information.

Even on ID inputs, a conditional mean need not follow the marginal Gaussian law of the injected noise. We therefore distinguish the hypothesis of a change in predicted-noise statistics from the stronger hypothesis that discrepancy from $\mathcal{N}(0, I)$ increases. The latter is evaluated empirically, rather than derived from reconstruction optimality. Algebraic connections to other parameterizations are discussed in Section C.

D.3 Posterior-Mean Variance

For a posterior-mean noise estimator, the law of total variance gives

$$\mathbb{E}\|\epsilon\|^2 = \mathbb{E}\|\mathbb{E}[\epsilon \mid x_t]\|^2 + \mathbb{E}\,\mathrm{tr}\,\mathrm{Var}(\epsilon \mid x_t). \tag{S7}$$

Thus its output second moment is at most d even on ID data. Gaussian injected noise does not make its conditional expectation a standard Gaussian.

D.4 Why OOD Does Not Imply a Universal Ordering

As a special case of the input shifts discussed in Section 3, consider replacing the data component by a fixed value while retaining Gaussian noise. A proposed ordering of the resulting output-moment statistics would be:

$$\mathbb{E}_{a,b}\Big[\|M^*(c_1 a + c_2 b)\|^2\Big] \le \mathbb{E}_a\Big[\|M^*(c_1 a + c_2 \tilde{b})\|^2\Big], \tag{S8}$$

where M^* denotes the Bayes-optimal estimator of a given the linear mixture $x = c_1 a + c_2 b$, b is drawn from the training distribution p_B, and $\tilde{b}$ is a fixed OOD sample. This proposed inequality would order the output second moments for these two conditions; it is not a consequence of Bayes optimality. Below, we clarify why this ordering requires additional assumptions.

Scope of the ordering hypothesis. Bayes optimality for squared reconstruction error gives the posterior mean $M^*(x) = \mathbb{E}[a \mid x]$, not a sample from the posterior. It does not by itself imply Eq. equation S8: reconstruction risk and the independent-reference output second moment are different quantities. Well-specification alone also does not guarantee that every distribution shift increases this statistic.

For example, let scalar a, b be independent standard normals. Then $M^*(x) = c_1 x/(c_1^2 + c_2^2)$ and, for a fixed replacement $\tilde{b}$,

$$R_{\mathrm{OOD}}(\tilde{b}) - R_{\mathrm{ID}} = \frac{c_1^2 c_2^2}{(c_1^2 + c_2^2)^2}(\tilde{b}^2 - 1).$$

The ordering holds for this example when $|\tilde{b}| \ge 1$ and reverses otherwise. Thus the relevant condition concerns the induced output moment, rather than an arbitrary ID/OOD label. Our experiments test the usefulness of the score for the evaluated shifts; they do not prove a universal typicality or OOD guarantee.

E Controlled Noise and OOD Diagnostics

These additional analyses separate sensitivity to noise statistics, ID/OOD discrimination, and downstream generation performance. They do not measure formal typical-set membership or causal mediation of preference. The revised generation selector uses only L2, KL, and squared Wasserstein. For continuity with the controlled diagnostic analysis, this section retains standard squared Hellinger as an auxiliary comparison statistic; it is not part of the revised selector.

Table S2: Noise-process ID/OOD AUROC from predicted-noise scores, with 128 paired images at $t = 250$. Every tested intervention is shown. Higher scores denote OOD without label-based sign reversal. L2 is the squared patch mean; energy is the full second moment. H^2 and Sum-4 are retained only as auxiliary diagnostic comparisons and are not part of the revised three-component generation selector.

Noise intervention	L2	Energy	KL	W_2^2	H^2	Sum-4
Mean shift $+0.1$	0.583	0.502	0.544	0.563	0.544	0.567
Mean shift $+0.3$	0.824	0.549	0.730	0.760	0.729	0.766
Residual SD $\times 0.5$	0.984	0.000	1.000	1.000	1.000	1.000
Residual SD $\times 0.8$	0.760	0.000	1.000	1.000	1.000	1.000
Residual SD $\times 1.2$	0.954	0.027	1.000	1.000	1.000	1.000
Residual SD $\times 2.0$	1.000	1.000	1.000	1.000	1.000	1.000
Sparse SD $\times 2$ (2/48 groups)	0.992	0.799	1.000	1.000	1.000	1.000
Sparse SD $\times 2$ (10/48 groups)	1.000	0.974	1.000	1.000	1.000	1.000
Moment-matched correlation	1.000	0.000	1.000	1.000	1.000	1.000
Moment-matched cubic tails	0.792	0.000	1.000	1.000	1.000	1.000
Energy-matched heterogeneous SD	0.996	0.000	1.000	1.000	1.000	1.000

E.1 Controlled Noise-Process Interventions

Controlled noise-process shifts. To separate the information captured by different statistics, we perturb forward noise while keeping the underlying clean image fixed. Using 128 held-out images and a disjoint class-matched clean calibration set, we preserve patch means while scaling within-patch standard deviations by $0.5, 0.8, 1.2, 2.0$. At the prespecified $t = 250$ and 4×4 grid, KL and squared Wasserstein each reach AUROC 1.000 on all four residual-scale interventions, demonstrating variance information beyond the squared-patch-mean baseline. The auxiliary squared-Hellinger and Sum-4 diagnostics are retained for comparison but are not used by the revised generation selector. Conversely, mean shifts favor the patch-mean statistic. Correlation and cubic-tail interventions rematched to the input patch means and variances are invisible to these input statistics, yet become separable in predicted-noise statistics after model evaluation. These constructed noise-process shifts provide a mechanism diagnostic; they are distinct from natural-image OOD detection and do not establish causal typicality-mediated improvements in generated-image preference.

We change injected forward noise while holding the clean image fixed. Patchwise moment matching is a synthetic control, and normalized cubic-Gaussian noise is not a camera-calibrated physical model. For the primary grid, each 256×256 prediction channel is split into 4×4 tiles with 4096 spatial values per tile. Moment preservation applies to the injected noise; the model can change output moments after inference.

For example, the cubic-tail intervention preserves input patch means and variances, leaving the listed input moment scores unchanged, yet changes predicted-noise variance and yields AUROC 1.000 for each variance-sensitive component at $t = 250$. This illustrates indirect sensitivity through the model, not direct tail detection by Gaussian moment formulas. A full-energy baseline and a squared-patch-mean baseline must also be distinguished: reverse ranking of raw energy can retain discriminative information, and ID calibration can exploit that information.

Table 1 in the main paper reports representative interventions; Table S2 and raw component scores. The additional analysis below asks whether clean-only calibration and multiple timesteps improve transfer to image-level distribution shifts; it is intentionally separate from the raw generation selector.

E.2 Broader ID/OOD Evaluation

Multi-statistic profiles across distribution shifts. We evaluate the three revised selector statistics (L2 mean-shift, Gaussian KL, and squared Wasserstein) together with auxiliary squared Hellinger on subsamples of ImageNet-A/R/O and eight ImageNet-C corruptions at three severities. A disjoint set of 192 clean images calibrates a fixed joint deviation score over these four diagnostic statistics at $t = 50, 250, 500$; no OOD labels are used for fitting. At the prespecified 4×4 grid, this detector attains a four-family macro AUROC of 0.601 (97.5% paired/independent bootstrap interval, as appropriate: $[0.565, 0.637]$), versus 0.524 ($[0.491, 0.558]$) for the raw four-statistic diagnostic sum

at $t = 250$. The interval level accounts for the two primary score comparisons. The calibrated detector reaches mean AUROC 0.658 on the 24 corruption conditions and 0.719 on ImageNet-R, but 0.472 on ImageNet-A and 0.557 on ImageNet-O. Its advantage over a calibrated L2-only multi-time detector is inconclusive (macro difference 0.007, pointwise 95% interval $[-0.018, 0.032]$). These results support limited, shift-dependent information in the divergence profile, rather than a universal raw-divergence ordering or a demonstrated need for the auxiliary Hellinger statistic. This clean-calibrated multi-time detector is distinct from the raw patch-score generation selector.

The evaluation uses 64 paired image identities for each of eight ImageNet-C corruptions at severities 1, 3, and 5, and 96 images each from ImageNet-A/R/O with clean controls. A/R controls match classes; O uses an independent ImageNet control set. The clean preprocessing approximates the ImageNet-C resize, crop, and JPEG construction, so this comparison is not an exact historical preprocessing reproduction. The joint score standardizes a 12-dimensional vector using clean calibration data and computes squared Mahalanobis distance with covariance shrinkage $0.9\widehat{\Sigma} + 0.1I$. It detects departures in either direction from the clean profile; it is not the raw sum used for candidate selection. Family averaging assigns equal weight to C, A, R, and O, with equal weighting of the 24 C conditions within that family.

Sensor noise and controlled statistical shifts. Real sensor noise motivates testing departures from a fixed Gaussian reference. Poissonian–Gaussian models capture signal-dependent variance in raw measurements Foi et al. (2008), and smartphone-camera data span noise levels that vary with capture conditions Abdelhamed et al. (2018). Quantile analysis finds sensor-noise tails heavier than conventional Poisson/Gaussian models predict Zhang et al. (2015); a low-light noise formation model uses a Tukey-lambda family to describe read-noise tails Wei et al. (2020). These observations motivate controlled variance and tail probes. Exact patch-mean preservation and moment-matched cubic noise in our experiments are isolation controls, not measured sensor laws. Moreover, observation noise in a camera and injected diffusion forward noise enter different processes. OOD status is always relative to a specified reference or training distribution.

F Additional Base-Model Results

F.1 Per-Model Estimates and Uncertainty

Table S3 collects the pointwise uncertainty estimates for all five metrics at both reported patch sizes, including the SANA unconditional-scoring control. Intervals use the prompt-bootstrap protocol in Appendix B and are relative to each setting's own vanilla pool.

F.2 Step and Patch Sweeps

For each base-model setting, the table reports HPSv2 changes at all five logged selection steps and six patch sizes. A patch size p denotes p^2 values per channel-wise group, not the number of groups. At $p = 1$ the within-group variance is zero, so the regularized score largely reduces to an energy ranking; this is a degenerate statistical control rather than a Gaussian fit with a well-estimated variance.

F.3 Compositional Controls and Sweeps

Table S5 includes the $p = 1$ control alongside the main reporting settings. For SANA-base under CFG++, $p = 1$ gives a Soft-TIFA gain of 5.33 points, compared with 4.96 at $p = 2$ and 4.71 at $p = 8$. Because singleton groups have zero within-group variance, this energy-like control prevents attributing the improvement specifically to local variance estimation. The score-component comparison above separately checks the effect of removing the Hellinger-like term. Table S6 retains every tested step and patch size for all four configurations, including negative cells. SANA CFG++ improves across all six patch sizes at $t_p = 15$, but several earlier and later large-patch cells are negative. Patch size therefore interacts with selection time; increasing it is not uniformly beneficial. The Bonferroni correction covers 30 cells within a configuration and metric, not comparisons across all backbones and metrics. All intervals condition on one generated pool per configuration.

Table S3: Base-model changes against each setting's own vanilla baseline at $t_p = 15$, with pointwise 95% paired prompt-bootstrap intervals. The two patch sizes are shown side by side.

Metric	$p = 2$: Δ [CI]	$p = 8$: Δ [CI]
SANA / CFG 4.5		
HPSv2	$+0.0019$ $[-0.0004, +0.0042]$	$+0.0035$ $[+0.0013, +0.0057]$
ImageReward	$+0.0116$ $[-0.0464, +0.0705]$	$+0.0386$ $[-0.0172, +0.0966]$
PickScore	$+0.1304$ $[+0.0604, +0.2048]$	$+0.1481$ $[+0.0761, +0.2201]$
CLIPScore	$+0.3104$ $[+0.0942, +0.5281]$	$+0.2384$ $[+0.0199, +0.4669]$
Aesthetic	$+0.0555$ $[+0.0201, +0.0910]$	$+0.0556$ $[+0.0198, +0.0904]$
SANA / CFG 4.5, unconditional score		
HPSv2	$+0.0062$ $[+0.0039, +0.0085]$	$+0.0072$ $[+0.0050, +0.0095]$
ImageReward	$+0.0894$ $[+0.0301, +0.1491]$	$+0.1003$ $[+0.0448, +0.1546]$
PickScore	$+0.2507$ $[+0.1808, +0.3236]$	$+0.2235$ $[+0.1531, +0.2954]$
CLIPScore	$+0.5996$ $[+0.3596, +0.8319]$	$+0.4704$ $[+0.2482, +0.6881]$
Aesthetic	$+0.1050$ $[+0.0704, +0.1405]$	$+0.1017$ $[+0.0667, +0.1366]$
SANA / CFG++ 0.6		
HPSv2	$+0.0100$ $[+0.0072, +0.0128]$	$+0.0132$ $[+0.0102, +0.0161]$
ImageReward	$+0.2133$ $[+0.1438, +0.2853]$	$+0.2513$ $[+0.1812, +0.3199]$
PickScore	$+0.3637$ $[+0.2751, +0.4546]$	$+0.3944$ $[+0.3043, +0.4816]$
CLIPScore	$+0.8974$ $[+0.6237, +1.1713]$	$+0.7378$ $[+0.4571, +1.0105]$
Aesthetic	$+0.1585$ $[+0.1178, +0.2010]$	$+0.1773$ $[+0.1315, +0.2194]$
SDXL / CFG 7.5		
HPSv2	$+0.0007$ $[-0.0009, +0.0022]$	$+0.0001$ $[-0.0014, +0.0017]$
ImageReward	$+0.0062$ $[-0.0347, +0.0477]$	$+0.0080$ $[-0.0288, +0.0462]$
PickScore	$+0.0045$ $[-0.0414, +0.0495]$	$+0.0136$ $[-0.0304, +0.0574]$
CLIPScore	-0.1110 $[-0.2658, +0.0391]$	$+0.0678$ $[-0.0935, +0.2250]$
Aesthetic	$+0.0018$ $[-0.0188, +0.0228]$	$+0.0061$ $[-0.0147, +0.0266]$
SDXL / CFG++ 0.6		
HPSv2	$+0.0025$ $[+0.0008, +0.0043]$	$+0.0010$ $[-0.0007, +0.0028]$
ImageReward	$+0.0268$ $[-0.0198, +0.0725]$	-0.0230 $[-0.0695, +0.0189]$
PickScore	$+0.0317$ $[-0.0216, +0.0814]$	$+0.0373$ $[-0.0103, +0.0855]$
CLIPScore	-0.0204 $[-0.1874, +0.1479]$	-0.0210 $[-0.2061, +0.1599]$
Aesthetic	$+0.0103$ $[-0.0144, +0.0352]$	$+0.0066$ $[-0.0191, +0.0335]$

Table S4: Complete base-model step/patch sweep: Δ HPSv2 relative to each setting's vanilla pool. These are exploratory effect sizes, without per-cell significance claims.

t_p	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$	$p = 32$
SANA / CFG 4.5						
5	+0.0032	+0.0026	-0.0004	+0.0002	-0.0013	-0.0013
10	-0.0012	+0.0009	-0.0014	+0.0007	+0.0006	+0.0015
15	+0.0030	+0.0019	+0.0004	+0.0035	+0.0045	+0.0064
20	-0.0015	-0.0010	+0.0044	+0.0057	+0.0075	+0.0080
25	-0.0036	+0.0020	+0.0038	+0.0031	+0.0030	+0.0031
SANA / CFG 4.5, unconditional score						
5	+0.0061	+0.0049	+0.0008	-0.0004	-0.0035	-0.0046
10	+0.0082	+0.0065	+0.0020	-0.0003	-0.0032	-0.0048
15	+0.0070	+0.0062	+0.0051	+0.0072	+0.0061	+0.0078
20	-0.0004	+0.0020	+0.0053	+0.0055	+0.0050	+0.0053
25	-0.0024	+0.0037	+0.0033	+0.0026	+0.0023	+0.0026
SANA / CFG++ 0.6						
5	+0.0084	+0.0044	-0.0001	+0.0002	+0.0004	-0.0020
10	+0.0130	+0.0106	+0.0034	-0.0013	-0.0059	-0.0077
15	+0.0111	+0.0100	+0.0125	+0.0132	+0.0126	+0.0118
20	+0.0021	+0.0018	+0.0034	+0.0067	+0.0057	+0.0052
25	-0.0014	+0.0061	+0.0049	+0.0045	+0.0046	+0.0051
SDXL / CFG 7.5						
5	-0.0007	+0.0002	-0.0023	-0.0006	+0.0000	+0.0011
10	-0.0008	+0.0002	-0.0009	-0.0000	+0.0009	-0.0003
15	+0.0017	+0.0007	-0.0006	+0.0001	-0.0013	-0.0023
20	+0.0018	+0.0021	+0.0007	-0.0012	-0.0034	-0.0035
25	+0.0018	+0.0038	-0.0003	-0.0010	-0.0030	-0.0030
SDXL / CFG++ 0.6						
5	-0.0016	-0.0010	-0.0014	+0.0009	+0.0027	+0.0022
10	+0.0002	+0.0014	+0.0003	+0.0018	+0.0028	+0.0013
15	+0.0011	+0.0025	+0.0002	+0.0010	+0.0001	-0.0023
20	+0.0007	+0.0021	+0.0003	-0.0002	-0.0015	-0.0031
25	+0.0002	+0.0028	+0.0017	-0.0019	-0.0028	-0.0027

Table S5: GenEval2 on the non-distilled backbones: 800 prompts, four candidates per prompt, $T = 30$, $t_p = 15$, predicted noise scored. Soft-TIFA is scaled by 100; every change is paired against the same-pool candidate mean; Oracle is the mean best-of-4 minus the candidate mean (the headroom). HPSv2 is measured on the same images. BoN-4 (IR) is the reward-based reference of the main table: it keeps the candidate with the highest ImageReward. Pointwise 95% bootstrap intervals are not adjusted for the scale sweep; † raw $p < .05$, ‡ Bonferroni over the 30 (t_p, p) cells (BoN rows: single test).

Model	Guidance	p	Vanilla Soft-TIFA	Oracle	Δ Soft-TIFA [CI]	Δ HPSv2 [CI]
SANA-base	CFG++ 0.6	1	40.73	+16.33	$+5.33^{\ddagger}$ [+4.39, +6.29]	$+0.0197^{\ddagger}$ [+0.0176, +0.0217]
SANA-base	CFG++ 0.6	2	40.73	+16.33	$+4.96^{\ddagger}$ [+4.05, +5.87]	$+0.0190^{\ddagger}$ [+0.0170, +0.0210]
SANA-base	CFG++ 0.6	8	40.73	+16.33	$+4.71^{\ddagger}$ [+3.77, +5.67]	$+0.0172^{\ddagger}$ [+0.0151, +0.0193]
SANA-base	CFG++ 0.6	BoN-4 (IR)	40.73	+16.33	$+9.59^{\ddagger}$ [+8.74, +10.44]	$+0.0298^{\ddagger}$ [+0.0283, +0.0315]
SANA-base	CFG 4.5	1	48.45	+15.12	$+2.42^{\ddagger}$ [+1.45, +3.39]	$+0.0107^{\ddagger}$ [+0.0087, +0.0126]
SANA-base	CFG 4.5	2	48.45	+15.12	$+1.47^{\dagger}$ [+0.52, +2.40]	$+0.0061^{\ddagger}$ [+0.0042, +0.0080]
SANA-base	CFG 4.5	8	48.45	+15.12	$+1.47^{\dagger}$ [+0.55, +2.37]	$+0.0080^{\ddagger}$ [+0.0060, +0.0099]
SANA-base	CFG 4.5	BoN-4 (IR)	48.45	+15.12	$+7.68^{\ddagger}$ [+6.82, +8.52]	$+0.0258^{\ddagger}$ [+0.0243, +0.0273]
SDXL-base	CFG++ 0.6	1	49.03	+13.11	−0.53 [−1.35, +0.29]	$-0.0019^{\dagger}$ [−0.0033, −0.0004]
SDXL-base	CFG++ 0.6	2	49.03	+13.11	$-0.86^{\dagger}$ [−1.68, −0.03]	−0.0003 [−0.0017, +0.0012]
SDXL-base	CFG++ 0.6	8	49.03	+13.11	$+0.91^{\dagger}$ [+0.08, +1.72]	$+0.0043^{\ddagger}$ [+0.0030, +0.0056]
SDXL-base	CFG++ 0.6	BoN-4 (IR)	49.03	+13.11	$+5.37^{\ddagger}$ [+4.61, +6.13]	$+0.0146^{\ddagger}$ [+0.0135, +0.0158]
SDXL-base	CFG 7.5	1	50.66	+13.20	+0.00 [−0.84, +0.83]	−0.0005 [−0.0020, +0.0009]
SDXL-base	CFG 7.5	2	50.66	+13.20	+0.07 [−0.74, +0.90]	+0.0007 [−0.0007, +0.0020]
SDXL-base	CFG 7.5	8	50.66	+13.20	−0.39 [−1.21, +0.43]	$+0.0023^{\ddagger}$ [+0.0010, +0.0035]
SDXL-base	CFG 7.5	BoN-4 (IR)	50.66	+13.20	$+5.17^{\ddagger}$ [+4.42, +5.99]	$+0.0127^{\ddagger}$ [+0.0116, +0.0139]

Table S6: Δ Soft-TIFA (×100, paired against the candidate mean) of the predicted-noise pick on the non-distilled backbones, over the pruning step t_p (rows, $T = 30$) and the patch size p (columns). † raw $p < .05$, ‡ Bonferroni over the 30 cells of each backbone.

t_p	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$	$p = 32$
SANA-base, CFG++ 0.6: vanilla 40.73, oracle +16.33*, P=800*						
5	$+2.62^{\ddagger}$	$+1.12^{\dagger}$	$+1.26^{\dagger}$	+0.19	$-1.13^{\dagger}$	−0.36
10	$+5.26^{\ddagger}$	$+4.18^{\ddagger}$	$+2.58^{\ddagger}$	−0.38	$-1.86^{\ddagger}$	$-1.37^{\dagger}$
15	$+5.33^{\ddagger}$	$+4.96^{\ddagger}$	$+4.70^{\ddagger}$	$+4.71^{\ddagger}$	$+4.77^{\ddagger}$	$+5.01^{\ddagger}$
20	$+3.23^{\ddagger}$	$+2.34^{\ddagger}$	$+1.16^{\dagger}$	−0.07	$-1.20^{\dagger}$	−0.91
25	$+1.63^{\ddagger}$	+0.06	+0.30	+0.18	+0.30	+0.32
SANA-base, CFG 4.5: vanilla 48.45, oracle +15.12*, P=800*						
5	$+2.03^{\ddagger}$	+0.41	+0.11	−0.43	$-1.81^{\ddagger}$	−0.57
10	$-1.13^{\dagger}$	−0.70	−0.23	−0.61	−0.91	−0.74
15	$+2.42^{\ddagger}$	$+1.47^{\dagger}$	+0.78	$+1.47^{\dagger}$	$+1.79^{\ddagger}$	$+2.20^{\ddagger}$
20	+0.55	−0.63	+0.61	$+1.75^{\ddagger}$	$+2.04^{\ddagger}$	$+1.90^{\ddagger}$
25	−0.77	$+1.15^{\dagger}$	$+1.64^{\ddagger}$	$+1.69^{\ddagger}$	$+1.92^{\ddagger}$	$+1.83^{\ddagger}$
SDXL-base, CFG++ 0.6: vanilla 49.03, oracle +13.11*, P=800*						
5	−0.15	−0.32	−0.42	+0.53	+0.47	$+1.05^{\dagger}$
10	−0.71	−0.37	+0.42	+0.78	+0.52	+0.77
15	−0.53	$-0.86^{\dagger}$	+0.32	$+0.91^{\dagger}$	+0.80	+0.41
20	−0.46	−0.15	+0.56	+0.76	+0.24	+0.55
25	−0.02	−0.33	+0.03	−0.30	−0.24	−0.07
SDXL-base, CFG 7.5: vanilla 50.66, oracle +13.20*, P=800*						
5	$-0.85^{\dagger}$	−0.65	−0.42	$-0.97^{\dagger}$	−0.79	$-1.25^{\dagger}$
10	−0.65	−0.40	−0.75	−0.19	−0.55	+0.15
15	+0.00	+0.07	−0.15	−0.39	−0.28	−0.00
20	+0.50	+0.13	+0.09	+0.01	−0.18	−0.18
25	+0.04	+0.17	+0.15	+0.16	−0.18	−0.06

F.4 Qualitative Examples

Figure S1 shows successful selections from the same four-candidate pools used in Table 2. Because the table's vanilla value is a candidate mean rather than a particular image, we use candidate 0 as a fixed vanilla sample and compare it with the candidate chosen by SelfVerify. The displayed prompts were selected after evaluation from cases with a positive HPSv2 difference, so they illustrate possible improvements rather than estimate their frequency or magnitude.

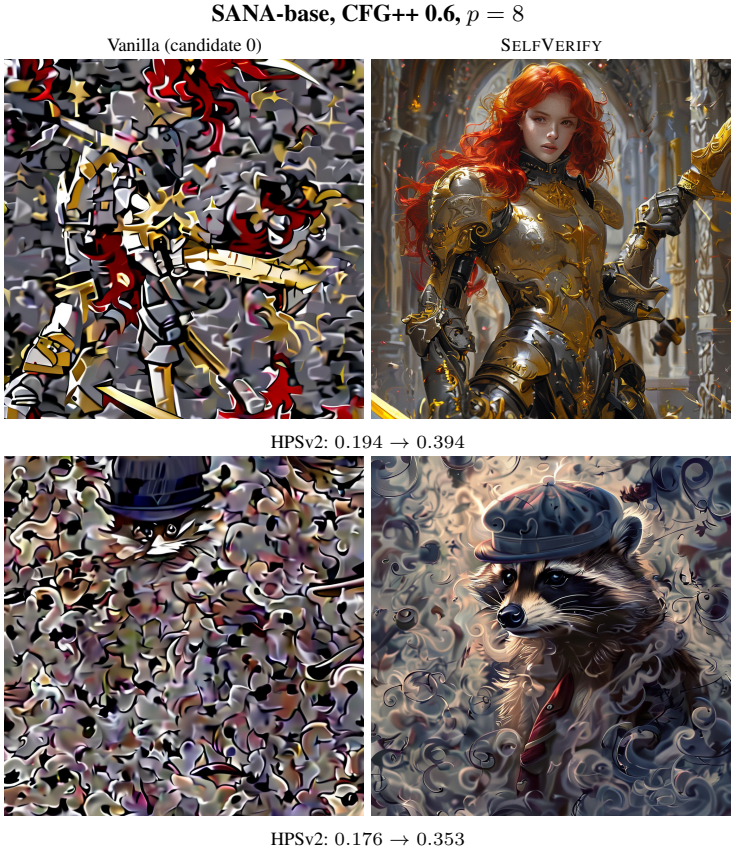

Figure S1: Qualitative examples from SANA-base CFG++ at $p = 8$. The prompts are "A red-haired female knight with a golden prosthetic arm wields a long golden blade" (top) and "A raccoon wearing a Peaky Blinders hat, surrounded by swirling mist" (bottom). HPSv2 is shown only for evaluation and was used to choose successful cases to display; these examples are not a random or representative sample.

F.5 Candidate-Count Surfaces

The candidate-count analysis is recomputed from the same SANA-base CFG++ 0.6 run using the revised three-term score. All cells use $p = 8$; subset identities are shared across steps, with 256 random subsets for $n < 16$ and the full pool for $n = 16$. Table S7 reports representative and late-step HPSv2 values from the recomputed surface. Heat-map significance uses the same Bonferroni correction over the 104 displayed step/count cells per metric.

G Transfer and Boundary Evaluations

G.1 Few-Step Distilled Backbones: A Boundary Evaluation

The main claims concern base models under guidance. For completeness, Table S8 reports all four completed HPDv2 evaluations on few-step distilled backbones. Here $n = 4$ is the candidate count, whereas $p \in \{2, 8\}$ is the spatial patch side length and corresponds to group size $M = p^2$.

Table S7: Representative and omitted late-step HPSv2 changes from the complete recorded surface. The final index 29 follows the last denoiser evaluation and offers no pruning work saving.

t_p	Target	$n = 2$	$n = 4$	$n = 8$	$n = 16$
15	eps	0.0066	0.0122	0.0164	0.0189
26	eps	0.0016	0.0033	0.0049	0.0058
27	eps	0.0020	0.0042	0.0061	0.0073
28	eps	0.0012	0.0030	0.0046	0.0054
29	eps	-0.0095	-0.0175	-0.0240	-0.0285

SDXL-Turbo is negative on HPSv2 at both displayed groupings, SANA-Sprint changes sign across groupings, Z-Image-Turbo is near zero, and FLUX-Schnell is positive only for selected metrics and groupings. These results do not support general transfer to distilled samplers. All scores use $\hat{\epsilon}$; SANA-Sprint and Z-Image use the corrected conversions.

Table S8: All four completed distilled HPDv2 evaluations, using predicted-noise selection and the corrected conversions. Every row uses $n = 4$ candidates; p is the spatial patch side length, so a group contains $M = p^2$ values within one channel. SDXL-Turbo uses $(T, t_p) = (1, 0)$, SANA-Sprint $(2, 1)$, Z-Image-Turbo $(8, 4)$, and FLUX-Schnell $(4, 2)$. † marks direct selection on ImageReward.

Selector	HPSv2	IR	PickScore	CLIP	Aesthetic
sdxl-turbo					
Vanilla	0.2848	0.8953	22.5915	29.4491	6.0384
SELFVERIFY $p = 2$	0.2798	0.9212	22.4835	29.5041	6.0165
SELFVERIFY $p = 8$	0.2799	0.9225	22.4863	29.4989	6.0175
BoN-4 (IR)	0.2901	1.1943†	22.7519	29.8342	6.0731
sana-sprint					
Vanilla	0.3025	1.0560	22.8374	28.6292	6.3753
SELFVERIFY $p = 2$	0.3049	1.0394	22.8169	28.5562	6.3584
SELFVERIFY $p = 8$	0.3003	1.0453	22.8127	28.6574	6.3699
BoN-4 (IR)	0.3070	1.3388†	22.9608	28.8415	6.3910
z-image-turbo					
Vanilla	0.2972	1.0199	22.3960	27.3077	5.8735
SELFVERIFY $p = 2$	0.2971	1.0344	22.4154	27.3228	5.8754
SELFVERIFY $p = 8$	0.2968	0.9983	22.3861	27.4213	5.8725
BoN-4 (IR)	0.3008	1.2233†	22.4885	27.6002	5.8816
flux-schnell					
Vanilla	0.3005	1.1267	22.5043	28.9764	6.1178
SELFVERIFY $p = 2$	0.2996	1.1363	22.5195	28.9348	6.1022
SELFVERIFY $p = 8$	0.3030	1.1332	22.5777	28.9427	6.1354
BoN-4 (IR)	0.3065	1.3999†	22.6388	29.2225	6.1452

G.2 AUDIO GROUPING AND SELECTION-TIME CONTROLS

UTMOS is the primary speech endpoint in this revision. The endpoint change is explicit; the current grids were available before the reporting choices and are not an independently preregistered test. The UTMOS $M = 1$ result is an energy-like control, not evidence that the variance-sensitive terms are necessary. Pointwise intervals in the scale/time sweep should not be read as multiplicity-corrected claims.

Table S9: AudioLDM2 predicted-noise selection: Δ UTMOS averaged over seeds 43–45 within each of 399 prompts. The full grid is exploratory; intervals are not adjusted for the 30 cells.

t_p	$M = 1$	$M = 4$	$M = 16$	$M = 64$	$M = 416$	$M = 1248$
5	+0.03469 [+0.02428, +0.04541]	+0.01438 [+0.00384, +0.02491]	+0.01134 [+0.00083, +0.02166]	+0.01472 [+0.00323, +0.02658]	+0.00647 [−0.00555, +0.01833]	+0.00642 [−0.00476, +0.01728]
12	+0.03592 [+0.02553, +0.04661]	+0.01690 [+0.00637, +0.02732]	+0.01237 [+0.00215, +0.02291]	+0.01505 [+0.00336, +0.02662]	+0.00814 [−0.00407, +0.02046]	+0.00807 [−0.00324, +0.01947]
25	+0.03599 [+0.02566, +0.04629]	+0.02763 [+0.01715, +0.03798]	+0.01392 [+0.00354, +0.02418]	+0.01786 [+0.00673, +0.02897]	−0.00474 [−0.01706, +0.00756]	+0.00011 [−0.01171, +0.01218]
37	+0.03545 [+0.02546, +0.04539]	+0.04108 [+0.03103, +0.05103]	+0.01818 [+0.00738, +0.02893]	+0.00565 [−0.00611, +0.01716]	−0.01443 [−0.02578, −0.00307]	−0.01633 [−0.02773, −0.00502]
45	+0.03607 [+0.02544, +0.04708]	+0.03794 [+0.02780, +0.04818]	+0.00650 [−0.00412, +0.01713]	−0.01543 [−0.02678, −0.00430]	−0.02896 [−0.04006, −0.01779]	−0.02799 [−0.03823, −0.01762]