LIVIN' ON A PRIOR: LIKELIHOOD SCORE APPROXIMATION FOR INVERSE PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models have found great success as data-driven methods of solving inverse problems. Two popular approaches work either by combining a pretrained generative prior with a known degradation model, or by training a conditional generative model directly from paired data. We target a setting that spans both regimes: unknown degradations can be learned from few paired examples, while known degradations can be learned from self-generated samples. We introduce *Likelihood Score Approximation (LSA)*, a generative framework that keeps a pretrained unconditional model fixed and learns an observation-conditioned model that approximates the likelihood score from paired samples. Within a conditional stochastic-interpolant framework, LSA can be trained in either score or velocity coordinates, independently of the unconditional model's native parameterization, and supports both deterministic and stochastic sampling. We further show empirically that the prior model can be swapped post-training while keeping the same LSA model. Across speech and image inverse problems, LSA operates effectively even at roughly 0.01% of the full training dataset. On the ImageNet-256 benchmark it achieves competitive or better restoration quality than strong posterior-sampling baselines while requiring up to several orders of magnitude fewer network evaluations.

1 INTRODUCTION

A wide range of problems in audio and imaging revolve around recovering a signal from degraded or incomplete data measurements. Examples of such tasks include super-resolution, inpainting, and deblurring in the image domain, as well as denoising, dereverberation, bandwidth extension, and long-audio inpainting in the audio domain (Kawar et al., 2022; Chung et al., 2023; Richter et al., 2023; Moliner et al., 2023; Welker et al., 2026). Many inverse problems can be described by the observation model

$$y = \mathcal{A}(x_0) + \eta, \tag{1}$$

where an unknown clean sample x_0 is corrupted by a measurement/degradation operator $\mathcal{A}$ plus additive measurement noise η, resulting in the measurement y. Since the observation model is generally non-invertible and/or corrupted by noise, reconstruction is often ill-posed and recovery relies on a prior over the signals of interest, to provide information that the observed data cannot.

Diffusion-based generative models have become a popular approach to tackling this class of problems. From a Bayesian perspective, the observation process is characterized by a likelihood $p(y \mid x_0)$. Together with a prior distribution $p(x_0)$ over clean data, this likelihood defines the posterior distribution $p(x_0 \mid y) \propto p(y \mid x_0)p(x_0)$ which maps the known measurement y to the unknown clean sample x_0. The Bayesian decomposition separates reusable knowledge of the clean-data distribution from observation-specific information encoded by the likelihood. In this view, different inverse problems may share the same prior over clean data, which can be represented by an unconditional generative model, while differing only in the likelihood induced by their respective observation process.

When the observation model $\mathcal{A}$ is known, a broad class of posterior-sampling methods combines a pretrained unconditional generative prior with the available measurement model directly during inference (Chung et al., 2023; Zhang et al., 2025a; Ahmed & Makin, 2026; Lemercier et al., 2025). Such methods require no task-specific paired training data, but rely on access to the observation model to construct observation-dependent guidance during sampling.

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.

When the observation model is unknown or unavailable, observation-specific information can instead be learned from paired clean–observation examples (x_0, y). Rather than constructing the posterior from an explicit measurement model at sampling time, conditional adaptation methods use these pairs to adapt a pretrained generative model to the inverse problem. When only a few paired examples are available, adapting a pretrained prior rather than training a conditional generative model from scratch is significantly more efficient. This adaptation can be done, for example, with ControlNet (Zhang et al., 2023) or parameter-efficient methods such as LoRA (Hu et al., 2022). ControlNet has been shown to remain trainable with limited conditional data, and Xu et al. (2025) use a ControlNet-based conditional score estimator trained on as few as 25 paired images for inverse problems.

This motivates a natural question: can we keep the clean-data prior fixed and learn only the observation-dependent component, using real paired data when the observation model is unavailable and self-generated pairs when it is known?

In this paper we introduce *Likelihood Score Approximation*, a framework in which a pretrained unconditional generative model is kept fixed and an observation-conditioned likelihood score approximation is learned from paired data. Unknown degradations can be learned from a few examples, while known degradation operators allow to work with only self-generated data. We show that the same framework can be applied to both score- and flow-based generative models and supports deterministic and stochastic sampling with test-time control. We evaluate LSA across speech and image inverse problems, including low-data, out-of-domain, and ImageNet 256×256 and 512×512 benchmarks. LSA also offers the possibility of prior swapping: the frozen prior can be replaced after training by another compatible generative model without retraining the LSA.

2 Likelihood Score Approximation with Stochastic Interpolants

Our goal is to extend a pretrained unconditional generative model (serving as a data prior) into an observation-conditioned posterior model. We achieve this by keeping the pretrained prior model fixed and approximating a likelihood score from paired data. In this section, we first review the Gaussian probability path and its score and velocity representations. We then introduce our method, derive the corresponding LSA training objectives, and finally describe deterministic and stochastic posterior samplers. Detailed derivations are provided in Appendix A.

2.1 Gaussian probability path, score and velocity

The generative models used in this work transform a simple noise distribution into the data distribution. We represent this transport through a Gaussian stochastic interpolant (Albergo et al., 2025), which defines a continuous path of intermediate distributions. This formulation places both unconditional and observation-conditioned generation within the same probability-path formulation. Let $X_0 \sim \pi_{\text{data}}$ denote a clean sample from the data distribution and let $\epsilon \sim \mathcal{N}(0, I)$ denote Gaussian noise independent of X_0. We consider the one-sided Gaussian stochastic interpolant (Albergo et al., 2025; Ma et al., 2024)

$$X_t = \alpha_t X_0 + \sigma_t \epsilon, \qquad t \in [0, 1], \tag{2}$$

where t denotes the interpolation time and X_t is the corresponding random state. The schedules α_t and σ_t are monotonically decreasing and increasing functions of t, respectively, controlling the interpolation between the clean sample and Gaussian noise. We choose $\alpha_0 = 1$, $\sigma_0 = 0$, and $\alpha_1 = 0$, such that $X_0 \sim \pi_{\text{data}}$ and $X_1 = \sigma_1 \epsilon \sim \mathcal{N}(0, \sigma_1^2 I)$.

Note that, throughout this paper, uppercase letters denote random variables and lowercase letters their realizations. For notational simplicity, we abbreviate $p_{X_t}(x)$, $p_{X_t|Y}(x \mid y)$, $p_{Y|X_t}(y \mid x)$, and $p_{X_t|X_0}(x \mid x_0)$ as $p_t(x)$, $p_t(x \mid y)$, $p_t(y \mid x)$, and $p_{0t}(x \mid x_0)$, respectively.

At each t, X_t has marginal density $p_t(x) = \int p_{0t}(x \mid x_0)\pi_{\text{data}}(x_0)\,\mathrm{d}x_0$, where $p_{0t}(x \mid x_0)$ is the conditional density induced by equation 2. The marginal density $p_t(x)$ is obtained by integrating out X_0. The resulting collection of densities $\{p_t\}_{t\in[0,1]}$ forms a continuous probability path between the data distribution and the Gaussian distribution.

Following the stochastic-interpolant formulation (Albergo et al., 2025), the probability path can be characterized by either of two vector fields: the score field $s(x, t)$, which encodes the geometry of

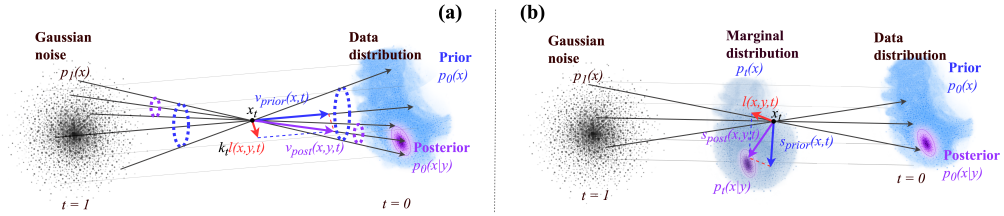

Figure 1: Prior and posterior vector fields along the Gaussian probability path. (a) Velocity vectors at the same intermediate state x_t, differing by $\kappa_t \ell_t(x, y)$. (b) Score vectors for $p_t(x)$ and $p_t(x \mid y)$, differing by $\ell_t(x, y)$.

the density $p_t(x)$, and the velocity field $v(x, t)$, which defines its transport over time:

$$s(x,t) := \nabla_x \log p_t(x) = \mathbb{E}\left[-\frac{\epsilon}{\sigma_t} \,\middle|\, X_t = x\right], \qquad v(x,t) := \mathbb{E}\left[\dot{X}_t \,\middle|\, X_t = x\right]. \tag{3}$$

Here, $\dot{X}_t = \dot{\alpha}_t X_0 + \dot{\sigma}_t \epsilon$ is the time derivative along an interpolant trajectory. Since different trajectories can pass through the same state x at time t, as illustrated in Fig. 1(a), $\dot{X}_t$ is not determined by $X_t = x$, and $v(x, t)$ is its conditional mean. The conditional-expectation expression for the score holds for $t > 0$, where $\sigma_t > 0$. These definitions follow the standard formulations of score-based diffusion (Song et al., 2021) and flow matching (Lipman et al., 2023). We refer to these unconditional vector fields $s(x, t)$ and $v(x, t)$ as the prior vector fields, denoted $s_{\text{prior}}(x, t)$ and $v_{\text{prior}}(x, t)$.

For $t \in (0, 1)$, the two fields are related explicitly (Albergo et al., 2025):

$$v(x,t) = \frac{\dot{\alpha}_t}{\alpha_t} x + \frac{\sigma_t(\dot{\alpha}_t \sigma_t - \alpha_t \dot{\sigma}_t)}{\alpha_t} s(x,t), \qquad s(x,t) = \frac{\alpha_t v(x,t) - \dot{\alpha}_t x}{\sigma_t(\dot{\alpha}_t \sigma_t - \alpha_t \dot{\sigma}_t)}. \tag{4}$$

These identities allow us to convert between score and velocity representations.

2.2 Observation-conditioned vector fields

In inverse problems, we seek the posterior distribution conditioned on the observation y introduced in equation 1, and accordingly condition the Gaussian probability path in equation 2 on y. Marginalizing over the clean samples X_0 under the clean-data posterior $p(x_0 \mid y)$ yields $p_t(x \mid y) = \int p_t(x \mid x_0) p(x_0 \mid y)\, \mathrm{d}x_0$, which describes the distribution of the perturbed state X_t at interpolation time t. Applying Bayes' rule then yields the score decomposition:

$$\underbrace{\nabla_x \log p_t(x \mid y)}_{\text{posterior score } s_{\text{post}}(x,y,t)} = \underbrace{\nabla_x \log p_t(x)}_{\text{prior score } s_{\text{prior}}(x,t)} + \underbrace{\nabla_x \log p_t(y \mid x)}_{\text{likelihood score } \ell_t(x,y)} \tag{5}$$

Here, $\ell_t(x, y)$ denotes the interpolation-time likelihood score. The interpolation-time likelihood $p_t(y \mid x)$ differs from the clean-data likelihood $p(y \mid x_0)$ because it is conditioned on the perturbed state $X_t = x$ rather than the clean sample $X_0 = x_0$, and is therefore generally intractable. A common line of posterior-sampling methods approximates this intractable term using the known degradation operator $\mathcal{A}$, typically through the clean-data likelihood $p(y \mid x_0)$ evaluated at an estimate of x_0 (Chung et al., 2023; Zhang et al., 2025a; Ahmed & Makin, 2026; Lemercier et al., 2025). In contrast, we propose to learn an approximation of the likelihood score from paired data.

Let $(X_0, Y) \sim \pi_{\text{pair}}$ denote paired clean samples and observations, and draw $\epsilon \sim \mathcal{N}(0, I)$. For a fixed interpolation time $t > 0$, we construct X_t using equation 2. Conditioning additionally on the observation Y induces the conditional probability path $p_t(x \mid y)$, whose data endpoint is the posterior $p(x_0 \mid y)$. Its corresponding score and velocity vector fields are (cf. equation 3):

$$s_{\text{post}}(x,y,t) = \mathbb{E}\left[-\frac{\epsilon}{\sigma_t} \,\middle|\, X_t = x, Y = y\right], \qquad v_{\text{post}}(x,y,t) = \mathbb{E}\left[\dot{X}_t \,\middle|\, X_t = x, Y = y\right]. \tag{6}$$

Importantly, conditioning does not change the sample-wise targets, which remain $-\epsilon/\sigma_t$ for the score and $\dot{X}_t$ for the velocity. Accordingly, Y enters only as an additional conditioning variable and does not appear in the regression target. This property was established for conditional score estimation by Batzolis et al. (2021) and for conditional velocity fields in flow matching by Dasgupta et al. (2026). Appendix A.1 gives the corresponding derivations in our stochastic-interpolant notation.

2.3 Decomposing posterior vector fields into prior and likelihood terms

As evident from the score decomposition in equation 5, the posterior and prior scores differ by the likelihood score. Combining this with the score–velocity relation in equation 4 gives the corresponding difference between their velocity fields:

$$\begin{aligned} s_{\text{post}}(x,y,t) - s_{\text{prior}}(x,t) &= \nabla_x \log p_t(y \mid x), \\ v_{\text{post}}(x,y,t) - v_{\text{prior}}(x,t) &= \kappa_t \, \nabla_x \log p_t(y \mid x), \end{aligned} \tag{7}$$

where $\kappa_t = \sigma_t(\dot{\alpha}_t\sigma_t - \alpha_t\dot{\sigma}_t)/\alpha_t$. Figure 1 illustrates the corresponding prior and posterior velocity vectors in panel (a) and score vectors in panel (b). Thus, the same likelihood score appears as $\ell_t(x,y)$ in score coordinates and as $\kappa_t\ell_t(x,y)$ in velocity coordinates. Besides the score and velocity formulations, the same decomposition can also be expressed in terms of directly predicting a clean-data estimate $\widehat{x}_0$, i.e., with a data prediction loss. This is useful for methods which require an explicit clean-sample estimate during sampling or training. The corresponding derivations are provided in Appendix A.2.

To turn this decomposition into a learnable model, we keep the pretrained prior fixed and learn the likelihood score approximation separately. Let $f \in \{s, v\}$ represent either the score (s) or velocity (v) parameterization of this likelihood term, and let $f_\theta(x,t)$ denote the output of the frozen prior model expressed in the corresponding parameterization (if the prior model uses a different native parameterization, its output must first be converted using equation 4). We then introduce the observation-conditioned LSA $\widehat{\ell}^f_\psi(x,y,t)$ and construct the posterior-field estimator

$$\widehat{f}_{\theta,\psi}(x,y,t) = f_\theta(x,t) + \widehat{\ell}^f_\psi(x,y,t), \qquad f \in \{s,v\}. \tag{8}$$

The prior depends only on (x,t), while LSA additionally receives the observation y. The two models share neither parameters nor intermediate features and are combined only by adding their outputs.

2.4 Training the Likelihood Score Approximation

We now turn the additive posterior model into a training objective for LSA. For that, we sample $(X_0, Y) \sim \pi_{\text{pair}}$, $\epsilon \sim \mathcal{N}(0, I)$, and $t \sim \mathcal{U}(t_{\text{eps}}, 1 - t_{\text{eps}})$, where $t_{\text{eps}} > 0$ excludes the path endpoints. We construct X_t using equation 2 and optimize

$$\mathcal{L}^f_{\text{LSA}}(\psi) = \mathbb{E}_{(X_0,Y),\,t,\,\epsilon}\left[\lambda_f(t)\left\|\text{sg}[f_\theta(X_t,t)] + \widehat{\ell}^f_\psi(X_t,Y,t) - T^f_t\right\|_2^2\right]. \tag{9}$$

Here, $\text{sg}[\cdot]$ denotes stop-gradient, so only the LSA parameters ψ are optimized, and $\lambda_f(t) > 0$ is an optional interpolation time-dependent weight. In practice, the expectation in $\mathcal{L}^f_{\text{LSA}}(\psi)$ is approximated by Monte Carlo averaging over sampled minibatches. We denote by T^f_t the corresponding sample-wise target in the chosen parameterization, with $T^s_t = -\epsilon/\sigma_t$ for the score and $T^v_t = \dot{X}_t$ for the velocity, as introduced in section 2.2. By applying equation 6, the conditional mean of these targets given $X_t = x$ and $Y = y$ is the corresponding posterior field $f_{\text{post}}(x,y,t)$.

Let $\widehat{\ell}^f(x,y,t)^*$ denote the optimum of the objective above. It therefore accounts for the difference between the frozen prior field and the posterior regression optimum. Hence,

$$\begin{aligned} \widehat{\ell}^f(x,y,t)^* &= f_{\text{post}}(x,y,t) - f_\theta(x,t) \\ &= \underbrace{f_{\text{post}}(x,y,t) - f_{\text{prior}}(x,t)}_{\ell^f_t(x,y)} + \underbrace{f_{\text{prior}}(x,t) - f_\theta(x,t)}_{\Delta^f_\theta(x,t)}. \end{aligned} \tag{10}$$

Here, ℓ^f_t denotes the likelihood score in the chosen parameterization, as defined in equation 7. The term Δ^f_θ captures the mismatch between the frozen prior model and the exact prior field. Thus, LSA can approximate the likelihood score up to the prior-model error Δ^f_θ. The corresponding regression derivation and the relation between the score and velocity LSA objectives are provided in Appendix A.3. Figure 2 summarizes the LSA training setup.

Prior swapping. The training procedure of LSA keeps the prior and LSA separate and combines them only through their output vector fields, allowing the prior to be swapped after training without retraining LSA. For example, if a better prior trained on more data becomes available or a specialized prior better matches the data distribution. The replacement prior must use the same α_t and σ_t as in equation 2. Since LSA is optimized relative to the original frozen prior, swapping the prior does not generally preserve the exact posterior field in equation 10. The efficacy of prior swapping is demonstrated in Section 5.1 and further analyzed in Appendix G.

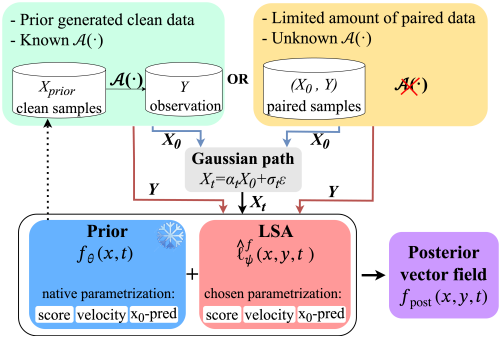

Figure 2: Overview of LSA. Training uses either self-generated pairs with a known degradation operator or limited paired-data with an unknown operator. Prior and LSA parameterizations can be chosen independently.

2.5 Deterministic and Stochastic Posterior Sampling

The frozen prior and LSA together define the posterior-field estimator in equation 8. The probability-flow ODE only requires the velocity field, while the stochastic-interpolant SDE additionally uses the score term. When only one representation $\widehat{f}_{\theta,\psi}(x,y,t) \in \{s, v\}$ is produced directly, the other is obtained from equation 4, without an additional model. Sampling starts from the Gaussian endpoint $X_1 \sim \mathcal{N}(0, \sigma_1^2 I)$, and proceeds toward the data endpoint. We use

$$\begin{aligned} \text{ODE:} \quad & dX_t = \widehat{v}_{\theta,\psi}(X_t, y, t)\, dt, \\ \text{SDE:} \quad & dX_t = \left[\widehat{v}_{\theta,\psi}(X_t, y, t) - \frac{1}{2} w_t \widehat{s}_{\theta,\psi}(X_t, y, t)\right] dt + \sqrt{w_t}\, d\overline{W}_t, \qquad w_t \geq 0. \end{aligned} \tag{11}$$

Here, $\overline{W}_t$ denotes a reverse-time Wiener process and w_t controls the sampler stochasticity. Setting $w_t = 0$ recovers the probability-flow ODE. With exact posterior fields, different choices of w_t define different sampling dynamics with the same posterior marginals $p_t(x \mid y)$ (Albergo et al., 2025).

3 Related Work

Learning residual corrections. The closest approach is DEFT (Denker et al., 2024), which likewise learns the posterior–prior score correction through a simulation-free denoising objective. The distinction is therefore in a way of representing and using the likelihood correction. In its reported inverse-problem implementations, DEFT conditions the learned h-transform on prior-derived features such as the unconditional Tweedie estimate and on operator-derived likelihood features; for inpainting, the network additionally receives the observation mask. Supervised Guidance Training (Baker et al., 2026) extends the DEFT principle to infinite-dimensional score-based diffusion models and also considering operator-informed likelihood features. In contrast, LSA receives only (x, y, t), requires neither prior outputs nor operator-derived features, and applies to both score and velocity parameterizations. Related flow-matching work also exploit additive conditional structure: CompFlow identifies the conditional–unconditional velocity residual with a noised likelihood gradient but uses it for training-free composition of conditional generators (Miglior et al., 2026).

Low-data conditional adaptation. Parameter-efficient adaptation of pretrained diffusion models has also been explored under limited supervision. ControlNet remains trainable with as few as 1k conditional pairs (Zhang et al., 2023), and Xu et al. (2025) use a ControlNet-based conditional score estimator trained on only 25 images for inverse problems. Low rank adaptation (LoRA) (Hu et al., 2022) adapts frozen pretrained models through low-rank weight updates; this strategy has also been applied directly to image restoration, with AdaptSR using LoRA to adapt pretrained super-resolution models to real-world degradations (Korkmaz et al., 2025). We therefore include ControlNet and LoRA as representative paired-data adaptation baselines in our low-data experiments.

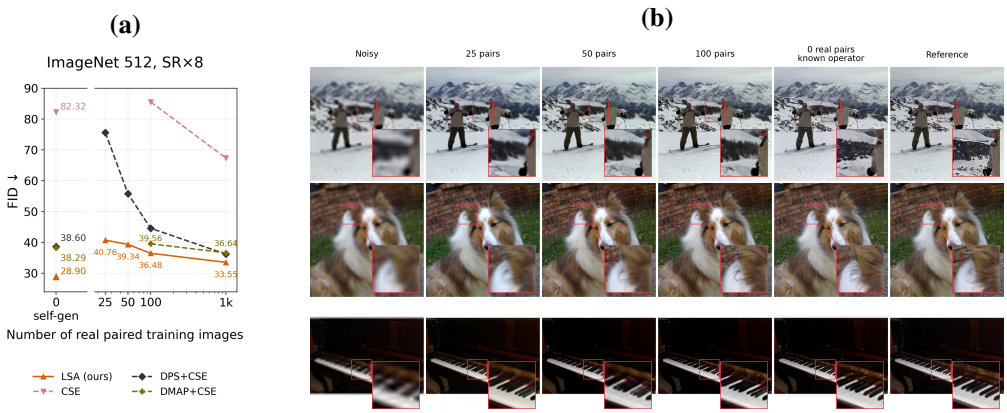

Figure 3: ImageNet-512 $8\times$ super-resolution across paired-data budgets, following the CSE benchmark (Xu et al., 2025). **(a)** FID for models trained with 25–1k real pairs, together with the self-generated setting. **(b)** Qualitative results for LSA.

4 Experimental setup

We evaluate LSA in two complementary regimes. First, we study whether a frozen generative prior can be adapted to an unknown observation process from only a small fixed set of paired examples. Second, when the degradation operator $\mathcal{A}$ is known, we show whether real paired data can be removed entirely by training the LSA on self-generated samples by the prior model. Across both regimes, we evaluate data efficiency, generality across inverse problems and modalities, and the quality-compute trade-off of the resulting posterior sampler.

We evaluate LSA on speech and image restoration tasks. For *speech*, we use clean speech recordings from the EARS dataset (Richter et al., 2024) and consider three inverse problems: speech enhancement, bandwidth extension, and short-time Fourier transform (STFT) phase retrieval. The frozen prior is trained on the full clean training split, whereas LSA without access to the degradation operator is trained from fixed paired subsets containing only 1 or 10 minutes of speech $\sim 0.16\%$ and 1.6% of EARS training data. The clean–observation pairs are sampled once and then held fixed: during low-data training, neither LSA nor the paired baselines access the degradation operator or resample observations online.

For ImageNet-256, we follow the DAPS benchmark (Zhang et al., 2025a) and evaluate $4\times$ super-resolution, box inpainting and random inpainting, Gaussian, motion, and nonlinear deblurring, and high-dynamic-range reconstruction. In the low-data setting, all paired methods use the same fixed set of 128 clean–observation pairs, $\sim 0.01\%$ of ImageNet training data, with a disjoint validation set used for model and sampler selection. Evaluation uses the released 100-image DAPS test subset.

We additionally consider a setting for known degradations. Here, clean training samples are generated from the frozen prior and passed through the known operator to construct self-generated pairs. The operator is therefore used to create the training observations, but no real paired examples are required. Finally, we evaluate $8\times$ super-resolution on ImageNet-512 following the low-data protocol of CSE (Xu et al., 2025), utilizing a randomly chosen 25, 50, 100, and 1,000 real paired images together with the self-generated setting.

Across all experiments, we use the linear Gaussian interpolant $\alpha_t = 1 - t$ and $\sigma_t = c\,t$, where $c = 1$ for image models and $c = \sigma_{\max}$ for speech to match the scale of the complex STFT representation. Appendix B specializes the score-velocity conversion and posterior ODE/SDE samplers to this path and provides all remaining training, sampling, and hyperparameter-selection details.

4.1 Models and baselines

For speech, the frozen prior is an unconditional NCSN++ (Song et al., 2021) score model trained on the full clean EARS training set. For the prior-swapping experiments, we additionally train prior models on the Urgent (Li et al., 2026) and VoiceBank (Valentini-Botinhao, 2017) datasets, as well as an ADM-based (Dhariwal & Nichol, 2021) prior on trained on the EARS. For images, we use

Table 1: Speech restoration results. (E), (U), and (VB) denote EARS, Urgent, and VoiceBank priors. *AnyEnhance is trained on 44.1-kHz audio. **Bold**: best within each data regime.

(a) Speech enhancement

Data	Model	PESQ ↑	SI-SDR ↑	Avg. nMOS ↑	FAD ↓
EARS-WHAM-v2					
1 min	Direct Posterior	1.44	8.70	2.60	1.95
	ControlNet	1.48	9.66	3.33	0.74
	LSA (ours)	**1.63**	**9.80**	**3.60**	**0.58**
10 min	Direct Posterior	1.70	11.50	3.44	0.25
	ControlNet	1.47	10.18	3.43	0.42
	LSA (ours)	**1.86**	**11.90**	**3.94**	**0.24**
Full	SBVE	2.15	12.90	4.09	1.05
	Direct Posterior	2.14	**14.75**	3.88	0.54
	ControlNet	1.63	12.62	3.49	0.46
	LSA (ours) (E)	2.21	14.35	**4.17**	**0.30**
	LSA (ours) (U)	**2.24**	14.16	4.06	0.50
VoiceBank Demand-ex (out-of-domain evaluation)					
Full	Direct Posterior	2.23	**13.22**	3.94	0.89
	LSA (ours) (E)	2.19	9.10	4.01	1.02
	LSA (ours) (U)	**2.32**	11.25	**4.07**	0.78
	LSA (ours) (VB)	**2.32**	9.87	4.05	**0.52**

(b) Bandwidth extension

Data	Model	PESQ ↑	LSD ↓	Avg. nMOS ↑	FAD ↓
1 min	Direct Posterior	1.81	1.40	3.43	0.38
	ControlNet	2.21	1.39	3.48	1.53
	LSA (ours)	**2.24**	**1.20**	**3.79**	**0.32**
10 min	Direct Posterior	2.42	1.11	3.86	0.20
	ControlNet	2.20	1.43	3.44	1.40
	LSA (ours)	**2.53**	**1.09**	**3.98**	**0.19**
Full	AnyEnhance*	2.50	1.40	4.12	0.64
	StreamFlow	**3.37**	1.26	3.78	0.71
	Direct Posterior	2.91	1.03	4.18	**0.11**
	ControlNet	2.27	1.20	3.83	0.49
	LSA (ours)	2.90	**0.99**	**4.20**	0.15

(c) STFT phase retrieval

Data	Model	PESQ ↑	LSD ↓	Avg. nMOS ↑	FAD ↓
1 min	ControlNet	**4.34**	0.81	4.07	0.079
	LSA (ours)	4.26	**0.42**	4.07	**0.059**
self-gen.	ControlNet	4.37	0.70	4.09	0.062
	LSA (ours)	**4.55**	**0.28**	**4.19**	**0.011**
Full	DiffPhase	4.04	0.53	4.10	0.046
	StreamFlow	4.24	0.76	4.20	0.106

pretrained PixelDiT[1] (Yu et al., 2026) models at 256×256 and 512×512 resolution and obtain the unconditional velocity field through their learned null condition. The prior remains frozen in all LSA experiments and shares neither parameters nor intermediate features with the LSA network.

For LSA, only the likelihood-score network is trained on the available paired data, using the objective in equation 9. For speech experiments we use the score parameterization, whereas the main image experiments use the velocity parameterization with the frozen PixelDiT velocity prior. Unless stated otherwise, we use an NCSN++ architecture for the LSA network. Architecture and training details are given in Appendix B.3.

In the paired low-data setting, we compare primarily against ControlNet and LoRA trained from exactly the same fixed pairs; for speech, we additionally train a direct conditional posterior model by training a single model from scratch with paired data. All auxiliary methods are trained to match a posterior vector field directly. We also evaluate DEFT (Denker et al., 2024) on the tasks for which a matched implementation is available. In the self-generated ImageNet-256 setting, we compare against established known-operator posterior samplers under the DAPS protocol. For ImageNet-512, we follow the low-data benchmark of Xu et al. (2025) and compare against the main configurations reported in that work.

4.2 Inference, hyperparameter selection, and evaluation

By default, we use deterministic posterior sampling with the probability-flow ODE, unless stated otherwise. For each task, we optimize the inference hyperparameters on held-out validation pairs and keep them fixed for test evaluation. The search covers the number of function evaluations and the two inference refinements described below. Sampling hyperparameters are optimized using PESQ/LSD for speech tasks and LPIPS for image tasks.

We additionally use two inference-time refinements to improve restoration quality. First, a guidance-like time-dependent scaling factor controls the contribution of the learned LSA relative to the frozen prior along the sampling trajectory. This allows us to vary the strength of the LSA model, analogously to guidance mechanisms used in diffusion models (Dhariwal & Nichol, 2021; Ho & Salimans, 2021). Second, we use a short restart-based refinement, motivated by restart-based generative and posterior sampling (Ahmed & Makin, 2026; Xu et al., 2023; Zhang et al., 2025a). After obtaining a restored estimate at the end point of the conditional trajectory, we perform a forward noising step to reintroduce noise at an intermediate point of the Gaussian path. From this restarted state, we integrate

[1] https://huggingface.co/nvidia/PixelDiT-ImageNet

Table 2: ImageNet-256 restoration on 100 images with additive Gaussian noise ($\sigma_y = 0.05$), following the DAPS benchmark (Zhang et al., 2025a). **Bold**: best. †DAPS rerun from DAPS++ (Chen et al., 2026).

Task	Method	NFE	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	Task	Method	NFE	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Super-resolution (4×)	DMAP	∼300	25.39	.661	.229	74.65	Gaussian deblurring	DAPS	1k	**26.15**	.684	.253	75.68
	DAPS	1k	**25.89**	.694	.276	83.57		ControlNet	26	25.08	.668	.242	99.82
	ControlNet	26	25.06	.675	.237	92.15		LSA (ours)	21	25.39	**.690**	**.212**	**69.34**
	LSA (ours)	23	25.57	**.700**	**.215**	**70.38**	Motion deblurring	RePS	1k	**28.95**	.801	.169	53.15
Inpainting (box)	MAS	–	21.15	**.817**	.168	95.96		DAPS†	1k	–	.769	.175	47.09
	DAPS	1k	**21.43**	.725	.214	109.85		LSA (ours)	42	28.18	**.802**	**.130**	**32.45**
	ControlNet	14	20.59	.683	.267	149.97	Nonlinear deblurring	RED-diff	1k	**30.07**	.754	.211	51.22
	LSA (ours)	34	20.42	.782	**.161**	**92.43**		DAPS	4k	27.73	.724	.169	59.87
Inpainting (random)	RePS	1k	**28.97**	.829	.115	27.52		LSA (ours)	18	27.10	**.789**	**.148**	**41.78**
	DAPS	1k	28.44	.775	.135	54.25	High dynamic range	RePS	4k	26.37	.843	.157	37.23
	LSA (ours)	12	28.75	**.847**	**.113**	**27.36**		DAPS	4k	26.30	.717	.175	64.19
								LSA (ours)	20	**28.30**	**.871**	**.120**	**25.64**

the conditional trajectory to the data endpoint again using the same observation y. Since the restart uses a newly sampled noise realization, the refinement follows a different conditional trajectory. By default, we perform a single restart unless stated otherwise. The definitions, interpretation, and hyperparameter optimization are detailed in Appendix B.4. All additional evaluations introduced by refinement are included in the reported NFE.

For image restoration, we report the default set used in inverse problems benchmarks: Peak signal-to-noise ratio (PSNR), SSIM Wang et al. (2004), LPIPS Zhang et al. (2018), and FID Heusel et al. (2017). For speech, the main tables show four complementary measures per task: PESQ Rix et al. (2001), DNN-based MOS prediction (as an average of SCOREQ Ragano et al. (2024), DistillMOS Stahl & Gamper (2025), and NISQA Mittag et al. (2021)), FAD Kilgour et al. (2019); Gui et al. (2024), and either SI-SDR Le Roux et al. (2019) for speech enhancement or log-spectral distance (LSD) for bandwidth extension and phase retrieval.

5 Results

5.1 Speech inverse problems

Evaluation results on the three speech restoration tasks are shown in Table 1. In the speech enhancement task, all models were trained on the EARS-WHAM-v2 dataset (Richter et al., 2024). The proposed LSA approach achieves considerably better results in the low-data scenario (1 min / 10 min of paired data) compared to both ControlNet and direct posterior training, especially w.r.t. FAD and DNN-based MOS scores. In the full-data regime, LSA still achieves competitive results, although the improvement upon direct posterior training is less pronounced in this case. Out-of-domain evaluation on the VoiceBank-DEMAND test set shows a certain degradation in performance. This degradation can be mitigated using prior-swapping: either by using a prior trained on a larger amount of data (U) or a prior trained on a smaller dataset which is more similar to the test data (VB). In both cases, only the pretrained prior model is swapped, while the LSA model is left unchanged (trained on the in-domain EARS-WHAM-v2).

Bandwidth extension shows the same low-data trend. With only 1 or 10 minutes of paired data, LSA achieves better results than both ControlNet and direct posterior training across all reported metrics. Figure 8 in Appendix F provides a qualitative spectrogram comparison across paired-data budgets. In the case of STFT phase retrieval, LSA achieves a reconstruction quality that is superior to other full-data approaches, even when using very limited paired data (1 min). In the self-generated data scenario with known corruption operator, LSA can harness the data generated by the simple forward operator and achieves an almost perfect reconstruction in terms of PESQ (4.55) and FAD (0.011).

5.2 Image inverse problems

Results for the tasks included in the DAPS benchmark (Zhang et al., 2025a) are presented in Table 2. LSA is compared against DAPS and, where applicable, against an additional current DPS-based baseline (chosen by best performance in terms of FID on each task).

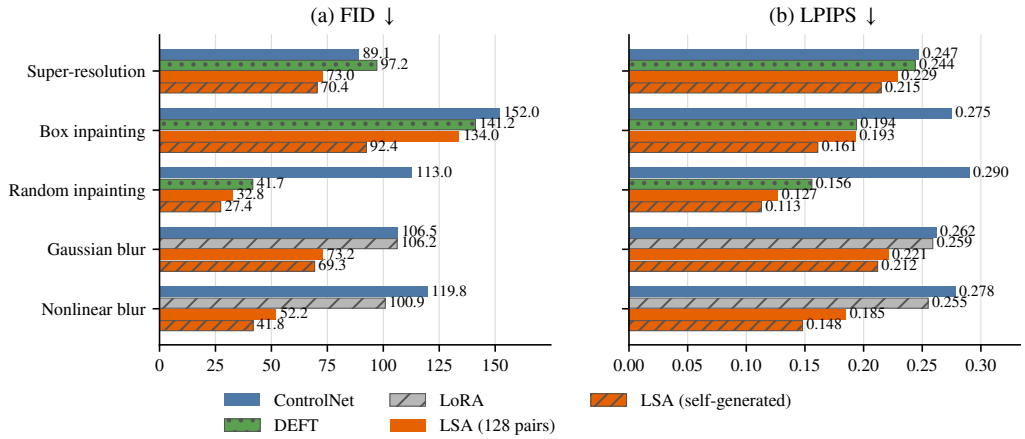

Figure 4: ImageNet-256 restoration with 128 real pairs. All paired methods are trained on the same fixed 128 pairs, with DEFT (Denker et al., 2024) included where available. For reference, we also show LSA trained on self-generated prior samples paired with observations from the known degradation operator.

These include DMAP (Xu et al., 2025), RePS (Ahmed & Makin, 2026), RED-diff (Mardani et al., 2024), and MAS (Zhang et al., 2025b). Compared to the competing approaches, LSA achieves considerably lower FID scores, as well as modestly better LPIPS and SSIM values and comparable PSNR. Note that all methods considered are prior-based and do not have access to paired data. However, the forward corruption operator is always known. Unlike the reported baselines, which are all based on the DPS paradigm and backpropagate through the forward operator, the conditional model obtained by LSA requires additional training using self-generated paired data derived from the known operator. Nevertheless, besides the improved reconstruction quality, this also allows for a drastic gain in inference efficiency: LSA requires fewer model evaluations; in some cases several orders of magnitude fewer than DPS-based approaches. Representative samples are shown in Figures 5–6 in Appendix C.

ImageNet-256, low-data regime. In Figure 4, we report FID and LPIPS scores on several image reconstruction tasks. LSA is compared here against ControlNet, LoRA and DEFT, all using 128 samples of paired data. In addition, self-generated results for LSA are included as a reference for the known-operator regime (see Table 2). LSA achieves strong reconstruction performance across all tasks and surpasses all low-data baselines in both metrics, in some cases approaching its self-generated known-operator counterpart. Full metrics for this comparison are presented in Table 6 in Appendix C.

ImageNet-512 8× super-resolution. For the specific task of 8× super-resolution on ImageNet-512 data, we evaluate LSA against CSE-based methods (Xu et al., 2025) for both the low-data regime and the self-generated known-operator case. In terms of FID (Figure 3a), LSA performs better than these methods on the self-generated data known-operator case and consistently well on the low-data regime, only requiring 100 data pairs for better reconstruction than the best-performing CSE method (DMAP+CSE) achieves with 1000 pairs. For a qualitative evaluation, we provide reconstructed images in Figure 3b for different amounts of paired data. Full comparison with CSE-based methods (Xu et al., 2025) can be found in Table 7 in Appendix D.

6 Conclusion

We introduced LSA, a generative framework for inverse problems that keeps a pretrained unconditional prior model fixed and approximates the observation-dependent likelihood score from paired data. Across speech and image restoration tasks, LSA remains effective in data-limited regimes, including settings with only minutes of paired speech or 25 paired ImageNet examples. When the degradation operator is known, the same framework can be trained from self-generated pairs and substantially reduce inference cost relative to diffusion posterior-sampling approaches. The same formulation applies to score and velocity formulations, supports deterministic and stochastic posterior sampling, and allows the prior model to be replaced at inference time without retraining LSA.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and local rewriting, literature retrieval and discovery, and limited coding support for debugging.

Generative AI tools were not used to formulate the research question, develop the core method or conceptual framework, formulate mathematical claims or derivations, provide proof arguments, propose hypotheses, design the experimental methodology, generate or process datasets, or interpret the reported results. The scientific ideas, methodology, mathematical development, experimental design, and conclusions were developed by the authors.

All AI-assisted text, literature suggestions, and code were reviewed and verified by the authors. References suggested by AI-assisted search were read by the authors, and AI-assisted code was tested before use. The authors take full responsibility for the final content of this work.

REFERENCES

Bilal Ahmed and Joseph G. Makin. Solving diffusion inverse problems with restart posterior sampling. In *European Conference on Computer Vision*, 2026. doi: 10.1007/978-3-032-37013-6_5.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26:1–80, 2025.

Elizabeth Louise Baker, Alexander Denker, and Jes Frellsen. Supervised guidance training for infinite-dimensional diffusion models. In *Int. Conf. Machine Learning (ICML)*, 2026.

Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021.

Hao Chen, Renzheng Zhang, and Scott S Howard. DAPS++: Rethinking diffusion inverse problems with decoupled posterior annealing. In *European Conference on Computer Vision*, pp. 223–240. Springer, 2026.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *Int. Conf. Learning Repr. (ICLR)*, 2023.

Agnimitra Dasgupta, Ali Fardisi, Mehrnegar Aminy, Brianna Binder, Bryan Shaddy, Saeed Moazami, and Assad A. Oberai. Conditional flow matching for physics-constrained inverse problems with finite training data. *arXiv preprint arXiv:2603.14135*, 2026.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alexander Denker, Francisco Vargas, Shreyas Padhy, Kieran Didi, Simon Mathis, Vincent Dutordoir, Riccardo Barbano, Emile Mathieu, Urszula Julia Komorowska, and Pietro Lio. DEFT: Efficient fine-tuning of diffusion models by learning the generalised h-transform. In *Adv. in Neural Information Proc. Sys.*, volume 37, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Adv. in Neural Information Proc. Sys.*, volume 34, pp. 8780–8794, 2021.

Azalea Gui, Hannes Gamper, Sebastian Braun, and Dimitra Emmanouilidou. Adapting frechet audio distance for generative music evaluation. In *IEEE Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP)*, pp. 1331–1335, 2024. doi: 10.1109/ICASSP48485.2024.10446663.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Adv. in Neural Information Proc. Sys.*, volume 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Int. Conf. Learning Repr. (ICLR)*, 2022.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Adv. in Neural Information Proc. Sys.*, volume 35, pp. 23593–23606, 2022. doi: 10.52202/068431-1714.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *ISCA Interspeech*, pp. 2350–2354, 2019. doi: 10.21437/Interspeech.2019-2219.

Cansu Korkmaz, Nancy Mehta, and Radu Timofte. Adaptsr: Low-rank adaptation for efficient and scalable real-world super-resolution. *arXiv preprint arXiv:2503.07748*, 2025.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey. SDR - Half-baked or well done? In *IEEE Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP)*, pp. 626–630, 2019. doi: 10.1109/ICASSP.2019.8683855.

Jean-Marie Lemercier, Eloi Moliner, Simon Welker, Vesa Välimäki, and Timo Gerkmann. Unsupervised blind joint dereverberation and room acoustics estimation with diffusion models. *IEEE/ACM Trans. Audio, Speech, Language Proc.*, 33:2244–2258, 2025. doi: 10.1109/TASLPRO.2025.3574988.

Chenda Li, Wei Wang, Marvin Sach, Wangyou Zhang, Kohei Saijo, Samuele Cornell, Yihui Fu, Zhaoheng Ni, Tim Fingscheidt, Shinji Watanabe, and Yanmin Qian. Icassp 2026 urgent speech enhancement challenge. In *IEEE Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP)*, pp. 21919–21921, 2026. doi: 10.1109/ICASSP55912.2026.11461686.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Int. Conf. Learning Repr. (ICLR)*, 2023.

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

Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *Int. Conf. Learning Repr. (ICLR)*, volume 2024, pp. 28027–28053, 2024.

Luca Miglior, Vincenzo Gervasi, and Davide Bacciu. CompFlow: Composing velocity fields for multi-condition generation. In *ICML Workshop on Compositional Learning: Safety, Interpretability, and Agents*, 2026.

Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller. NISQA: A deep CNN-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. In *ISCA Interspeech*, pp. 2127–2131, 2021. doi: 10.21437/Interspeech.2021-299.

Eloi Moliner, Jaakko Lehtinen, and Vesa Välimäki. Solving audio inverse problems with a diffusion model. In *IEEE Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10095637.

Alessandro Ragano, Jan Skoglund, and Andrew Hines. SCOREQ: Speech quality assessment with contrastive regression. In *Adv. in Neural Information Proc. Sys.*, volume 37, pp. 105702–105729, 2024. doi: 10.52202/079017-3353.

Julius Richter, Simon Welker, Jean-Marie Lemercier, Bunlong Lay, and Timo Gerkmann. Speech enhancement and dereverberation with diffusion-based generative models. *IEEE/ACM Trans. Audio, Speech, Language Proc.*, 31:2351–2364, 2023.

Julius Richter, Yi-Chiao Wu, Steven Krenn, Simon Welker, Bunlong Lay, Shinji Watanabe, Alexander Richard, and Timo Gerkmann. EARS: An anechoic fullband speech dataset benchmarked for speech enhancement and dereverberation. In *ISCA Interspeech*, pp. 4873–4877, 2024. doi: 10.21437/Interspeech.2024-153.

A.W. Rix, J.G. Beerends, M.P. Hollier, and A.P. Hekstra. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *IEEE Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP)*, volume 2, pp. 749–752 vol.2, 2001. doi: 10.1109/ICASSP.2001.941023.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Int. Conf. Learning Repr. (ICLR)*, 2021.

Benjamin Stahl and Hannes Gamper. Distillation and pruning for scalable self-supervised representation-based speech quality assessment. In *IEEE Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10888007.

Cassia Valentini-Botinhao. Noisy speech database for training speech enhancement algorithms and tts models, 2017. URL `https://doi.org/10.7488/ds/2117`.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Proc.*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Simon Welker, Bunlong Lay, Maris Hillemann, Tal Peer, and Timo Gerkmann. Real-time streamable generative speech restoration with flow matching. *IEEE/ACM Trans. Audio, Speech, Language Proc.*, 34:3087–3101, 2026. doi: 10.1109/TASLPRO.2026.3696215.

Tongda Xu, Xiyan Cai, Xinjie Zhang, Xingtong Ge, Dailan He, Ming Sun, Jingjing Liu, Ya-Qin Zhang, Jian Li, and Yan Wang. Rethinking diffusion posterior sampling: From conditional score estimator to maximizing a posterior. In *Int. Conf. Learning Repr. (ICLR)*, 2025.

Yilun Xu, Mingyang Deng, Xiang Cheng, Yonglong Tian, Ziming Liu, and Tommi Jaakkola. Restart sampling for improving generative processes. In *Adv. in Neural Information Proc. Sys.*, volume 36, 2023.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. pp. 14273–14282, 2026.

Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2025a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pp. 3836–3847, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

Shaorong Zhang, Rob Brekelmans, Yunshu Wu, and Greg Ver Steeg. Measurement-aligned sampling for inverse problem, 2025b.

A CONDITIONAL VECTOR FIELDS AND LSA REGRESSION

This appendix collects the derivations underlying the conditional regression, score–velocity conversion and LSA objectives.

A.1 CONDITIONAL REGRESSION FROM PAIRED SAMPLES

This subsection justifies the conditional field identities in equation 6. For paired samples $(X_0, Y) \sim \pi_{\text{pair}}$, conditioning on a fixed observation $Y = y$ changes the clean endpoint distribution to $p(x_0 \mid y)$, while the Gaussian interpolant and its sample-wise score and velocity targets remain unchanged. We show below that these targets correspond to the posterior score and velocity fields of the conditional probability path $p_t(x \mid y)$.

For the Gaussian interpolant in equation 2, the conditional marginal at time t is

$$p_t(x \mid y) = \int p_{0t}(x \mid x_0) p(x_0 \mid y)\, dx_0, \qquad p_{0t}(x \mid x_0) = \mathcal{N}(x; \alpha_t x_0, \sigma_t^2 I). \tag{12}$$

Differentiating the logarithm of the conditional marginal gives

$$\begin{aligned}
s_{\text{post}}(x, y, t) &:= \nabla_x \log p_t(x \mid y) && (13)\\
&= \frac{1}{p_t(x \mid y)} \int p(x_0 \mid y) \nabla_x p_t(x \mid x_0, y)\, dx_0 && (14)\\
&= \int \frac{p_t(x \mid x_0, y) p(x_0 \mid y)}{p_t(x \mid y)} \nabla_x \log p_t(x \mid x_0, y)\, dx_0 && (15)\\
&= \int p(x_0 \mid x, y) \nabla_x \log p_t(x \mid x_0, y)\, dx_0 && (16)\\
&= \int p(x_0 \mid x, y) \nabla_x \log p_{0t}(x \mid x_0)\, dx_0 && (17)\\
&= \mathbb{E}[\nabla_x \log p_{0t}(x \mid x_0) \mid X_t = x, Y = y]\,. && (18)
\end{aligned}$$

The transition from line (16) to line (17) follows from the construction of the Gaussian interpolant. After conditioning on $X_0 = x_0$, the perturbed state is generated as $X_t = \alpha_t x_0 + \sigma_t \epsilon$, where ϵ is independent of Y. Since ϵ is sampled independently of (X_0, Y), we have $X_t \perp Y \mid X_0$ and

$$p_t(x \mid x_0, y) = p_{0t}(x \mid x_0). \tag{19}$$

For the Gaussian transition,

$$\nabla_x \log p_{0t}(x \mid x_0) = -\frac{x - \alpha_t x_0}{\sigma_t^2} = -\frac{\epsilon}{\sigma_t}, \tag{20}$$

and therefore

$$s_{\text{post}}(x, y, t) = \mathbb{E}\left[-\frac{\epsilon}{\sigma_t} \,\middle|\, X_t = x, Y = y \right]. \tag{21}$$

For flow matching, the sample-wise target remains

$$T_t^{\text{v}} = \dot{X}_t = \dot{\alpha}_t X_0 + \dot{\sigma}_t \epsilon. \tag{22}$$

To obtain the corresponding posterior velocity field, consider a fixed observation $Y = y$. Conditioning on $Y = y$ restricts the clean endpoint to $X_0 \sim p(x_0 \mid y)$, while the interpolant $X_t = \alpha_t X_0 + \sigma_t \epsilon$ and its sample-wise velocity $\dot{X}_t$ remain unchanged. The resulting conditional probability path is $p_t(x \mid y)$, whose data endpoint is the posterior $p(x_0 \mid y)$. Applying the same stochastic-interpolant velocity identity as in equation 3 to this conditional distribution therefore gives

$$v_{\text{post}}(x, y, t) = \mathbb{E}\left[\dot{X}_t \,\middle|\, X_t = x, Y = y \right]. \tag{23}$$

Thus, v_{post} is the velocity field associated with the conditional probability path $p_t(x \mid y)$: it averages the same sample-wise velocities $\dot{X}_t$, but only over trajectories compatible with both $X_t = x$ and $Y = y$. For a derivation showing that this velocity field transports the corresponding conditional probability path, see Dasgupta et al. (2026, Proposition 2.1 and Section 2.2).

A.2 SCORE-VELOCITY CONVERSION AND POSTERIOR-PRIOR DECOMPOSITION

Let $\mathcal{C} = X_t = x$ for the prior field and $\mathcal{C} = (X_t = x, Y = y)$ for the posterior field. In either case, the Gaussian path gives

$$x = \alpha_t \mathbb{E}[X_0 \mid \mathcal{C}] + \sigma_t \mathbb{E}[\epsilon \mid \mathcal{C}], \qquad v(x,t) = \dot{\alpha}_t \mathbb{E}[X_0 \mid \mathcal{C}] + \dot{\sigma}_t \mathbb{E}[\epsilon \mid \mathcal{C}], \tag{24}$$

with

$$\mathbb{E}[\epsilon \mid \mathcal{C}] = -\sigma_t s(x,t).$$

Eliminating the conditional clean mean gives the score–velocity relation in equation 4 for either choice of $\mathcal{C}$. Hence,

$$\begin{aligned} v_{\text{prior}}(x,t) &= \frac{\dot{\alpha}_t}{\alpha_t} x + \kappa_t s_{\text{prior}}(x,t), \\ v_{\text{post}}(x,y,t) &= \frac{\dot{\alpha}_t}{\alpha_t} x + \kappa_t s_{\text{post}}(x,y,t), \end{aligned} \qquad \kappa_t := \frac{\sigma_t\big(\dot{\alpha}_t \sigma_t - \alpha_t \dot{\sigma}_t\big)}{\alpha_t}. \tag{25}$$

By Bayes' rule,

$$s_{\text{post}}(x,y,t) - s_{\text{prior}}(x,t) = \nabla_x \log p_t(y \mid x) =: \ell_t(x,y). \tag{26}$$

Subtracting the two instances of equation 25 cancels the common term $\frac{\dot{\alpha}_t}{\alpha_t} x$, giving

$$\begin{aligned} v_{\text{post}}(x,y,t) - v_{\text{prior}}(x,t) &= \kappa_t \left(s_{\text{post}}(x,y,t) - s_{\text{prior}}(x,t)\right) \\ &= \kappa_t \ell_t(x,y). \end{aligned} \tag{27}$$

Thus, the posterior-prior difference carries the same likelihood score in score and velocity coordinates, but scaled by κ_t for the velocity. The same conditional–unconditional velocity residual was derived by Miglior et al. (2026) for a specific Gaussian probability path.

Data prediction formulation. The same conditional probability path can also be represented through the conditional mean of the clean-data X_0. We define the prior and posterior data-prediction fields as

$$d_{\text{prior}}(x,t) := \mathbb{E}[X_0 \mid X_t = x], \qquad d_{\text{post}}(x,y,t) := \mathbb{E}[X_0 \mid X_t = x, Y = y]. \tag{28}$$

Unlike the velocity field, these quantities do not themselves define the transport of the probability path. For the Gaussian interpolant, however, they are exactly related to the corresponding score and velocity fields. For either the prior or posterior conditioning, using equation 2 gives

$$d(x,t) = \frac{x + \sigma_t^2 s(x,t)}{\alpha_t} = \frac{\sigma_t v(x,t) - \dot{\sigma}_t x}{\dot{\alpha}_t \sigma_t - \alpha_t \dot{\sigma}_t}. \tag{29}$$

These relations correspond to the one-sided stochastic-interpolant identities in Albergo et al. (2025, Section 4.4, Eqs. (4.16)–(4.18)). Thus, the same data prediction can be obtained from either the score or velocity field.

Again, applying equation 29 to the prior and posterior fields and subtracting gives

$$\begin{aligned} d_{\text{post}}(x,y,t) - d_{\text{prior}}(x,t) &= \frac{\sigma_t^2}{\alpha_t} \left(s_{\text{post}}(x,y,t) - s_{\text{prior}}(x,t)\right) \\ &= \frac{\sigma_t^2}{\alpha_t} \ell_t(x,y). \end{aligned} \tag{30}$$

Thus, as in the velocity posterior-prior difference equation 27, the posterior-prior difference in data-prediction coordinates is proportional to the same likelihood score $\ell_t(x,y)$, up to a known path-dependent factor σ_t^2/α_t.

A.3 LSA REGRESSION AND PARAMETERIZATION EQUIVALENCE

Choose the LSA parameterization $f \in \{s, v\}$ and express the frozen prior output as f_θ using equation 3. The assembled predictor is

$$\widehat{f}_{\theta,\psi}(x, y, t) = f_\theta(x, t) + \widehat{\ell}^f_\psi(x, y, t) \tag{31}$$

Here, f_θ remains fixed during training, regardless of the prior's native parameterization. We consider squared-error regression of the assembled predictor toward the sample-wise target T^f_t, while keeping f_θ fixed. At the population level, assuming an unrestricted LSA predictor, i.e., one that can represent any square-integrable function of (x, y, t), optimizing the LSA model is equivalent to optimizing the assembled predictor. By the standard squared-error regression identity, the population optimum of the assembled predictor is

$$\widehat{f}^*_{\theta,\psi}(x, y, t) = \mathbb{E}\Big[T^f_t \;\Big|\; X_t = x, Y = y\Big] = f_{\text{post}}(x, y, t) \tag{32}$$

Thus, the assembled model has the same population optimum as ordinary conditional score or flow regression.

Since f_θ is fixed, the optimal likelihood score approximation is obtained by subtracting the frozen field:

$$\widehat{\ell}^*_f(x, y, t) = f_{\text{post}}(x, y, t) - f_\theta(x, t) \tag{33}$$

Thus, at the unrestricted population optimum, LSA recovers the difference between the posterior field and the frozen prior model.

A time-weighted squared-error objective that achieves this population optimum for both parameterizations is

$$\mathcal{L}^f_{\text{LSA}}(\psi) = \mathbb{E}_{\substack{(X_0,Y)\sim\pi_{\text{pair}} \\ t\sim\mathcal{U}(t_{\text{eps}},1-t_{\text{eps}}) \\ \epsilon\sim\mathcal{N}(0,I)}} \left[\lambda_f(t) \left\| f_\theta(X_t, t) + \widehat{\ell}^f_\psi(X_t, Y, t) - T^f_t \right\|_2^2\right] \tag{34}$$

Since $\lambda_f(t) > 0$ depends only on t, the weighting does not change the pointwise population optimum derived above. The prior output, likelihood score approximation, and sample-wise target are all expressed in the chosen parameterization f.

The two forms are connected by the same affine score–velocity relation derived above. Using $X_t = \alpha_t X_0 + \sigma_t \epsilon$ to eliminate X_0 gives

$$T^v_t = \frac{\dot{\alpha}_t}{\alpha_t} X_t + \kappa_t T^s_t. \tag{35}$$

To compare the two objectives, we express the same frozen prior in score and velocity coordinates and use $\widehat{\ell}^v_\psi = \kappa_t \widehat{\ell}^s_\psi$. The assembled fields then satisfy:

$$\widehat{v}_{\theta,\psi} = \frac{\dot{\alpha}_t}{\alpha_t} X_t + \kappa_t \widehat{s}_{\theta,\psi}. \tag{36}$$

Subtracting the corresponding targets cancels the common affine term, yielding

$$\|\widehat{v}_{\theta,\psi} - T^v_t\|_2^2 = \kappa_t^2 \, \|\widehat{s}_{\theta,\psi} - T^s_t\|_2^2 \,. \tag{37}$$

For these compatible representations, the unweighted losses at fixed time differ by κ_t^2, and their population optima correspond under the same conversion. Choosing $\lambda_s(t) = \kappa_t^2 \lambda_v(t)$ makes the time-weighted objectives identical.

Data-prediction parameterization. The same residual-regression construction applies in data-prediction coordinates, using the sample-wise target $T^d_t = X_0$. Accordingly, the corresponding LSA objective is obtained from equation 34 by replacing f_θ, $\widehat{\ell}^f_\psi$, and T^f_t with d_θ, $\widehat{\ell}^d_\psi$, and X_0, respectively.

B EXPERIMENTAL DETAILS

B.1 LINEAR GAUSSIAN PROBABILITY PATH

Throughout the experiments, we use the linear Gaussian path

$$X_t = (1-t)X_0 + ct\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \qquad t \in [0,1], \tag{38}$$

corresponding to $\alpha_t = 1 - t$ and $\sigma_t = ct$. We use $c = 1$ for image experiments and $c = \sigma_{\max} = 0.5$ for speech, where the latter accounts for the scale of the compressed complex-STFT representation.

Substituting these schedules into Eq. equation 4 gives the direct conversion between velocity and score fields:

$$v(x,t) = -\frac{x + c^2 t\, s(x,t)}{1-t}, \qquad s(x,t) = -\frac{(1-t)v(x,t) + x}{c^2 t}. \tag{39}$$

Thus, trained score or velocity predictions can be converted analytically into the representation required by the sampler without an additional network.

For this path, the sample-wise LSA training targets from Eq. equation 9 reduce to

$$T_t^s = -\frac{\epsilon}{ct}, \qquad T_t^v = -X_0 + c\epsilon. \tag{40}$$

Stochastic sampling We additionally study stochastic posterior sampling by varying the diffusion coefficient w_t in equation 11. Following Ma et al. (2024), we consider the KL-motivated diffusion coefficient

$$w_t^{\mathrm{KL}} = 2\left(\dot{\sigma}_t \sigma_t - \frac{\dot{\alpha}_t \sigma_t^2}{\alpha_t}\right), \tag{41}$$

which minimizes an upper bound on the terminal KL divergence when numerical integration cost is disregarded. This choice also coincides with the variance rate of the linear Markov diffusion whose perturbation kernel has the same (α_t, σ_t). Consequently, $w_t = w_t^{\mathrm{KL}}$ recovers the corresponding conventional reverse-time SDE (Song et al., 2021).

For the linear Gaussian path used in our experiments, $\alpha_t = 1 - t$ and $\sigma_t = ct$, the diffusion coefficient is:

$$w_t^{\mathrm{KL}} = \frac{2c^2 t}{1-t}. \tag{42}$$

Since we are free to use an arbitrary diffusion coefficient (Albergo et al., 2025), we additionally introduce a constant overall stochasticity level ρ:

$$w_t = \rho\, w_t^{\mathrm{KL}}, \qquad \rho \geq 0, \tag{43}$$

where $\rho = 0$ recovers deterministic sampling and $\rho = 1$ recovers the KL-motivated reverse-SDE coefficient.

B.2 DATASETS AND OBSERVATION PROTOCOLS

Speech restoration. We use EARS (Richter et al., 2024) as the clean-speech dataset and consider three restoration tasks: speech enhancement, bandwidth extension, and STFT phase retrieval. Speech enhancement follows the EARS-WHAM v2 protocol, while bandwidth extension and phase retrieval follow the task definitions of Welker et al. (2026). For the low-data experiments, we construct nested fixed subsets containing 1 and 10 minutes of paired data, corresponding to 6 and 60 ten-second pieces, respectively. The clean examples and their observations are generated once and kept fixed throughout training for all paired methods.

For bandwidth extension, observations remain at 16 kHz but are low-pass filtered with a cutoff randomly selected between 2 and 4 kHz, following the reference protocol. For phase retrieval, we use the STFT-based observation defined by Welker et al. (2026). Hyperparameter selection uses the EARS validation split, and final evaluation uses the corresponding test split. We additionally evaluate speech enhancement out-of-domain on VoiceBank-DEMAND extended [2] while training LSA on EARS-WHAM pairs.

[2] https://www.emergentmind.com/topics/voicebank-demand-extended-vb-demandex

ImageNet-256. For ImageNet-256, we follow the public DAPS benchmark (Zhang et al., 2025a). Evaluation uses the released subset of 100 ImageNet images (Deng et al., 2009), and for each task we reproduce the corresponding observation using the forward operators and corruption parameters provided with the benchmark[3]. We consider $4\times$ super-resolution, box and random inpainting, Gaussian, motion, and nonlinear deblurring, and high-dynamic-range reconstruction.

In the paired low-data setting, all methods use the same fixed subset of 128 ImageNet training images and one fixed observation per image. We additionally sample 25 non-intersecting ImageNet training images for validation. The clean training subset is shared across tasks, while observations are constructed separately for each degradation. Neither the degradation operator nor newly generated observations are available during low-data training.

ImageNet-512 $8\times$ super-resolution. For ImageNet-512, we follow the $8\times$ super-resolution protocol of Xu et al. (2025). We evaluate paired-data budgets of 25, 50, 100, and 1,000 training images. The 25-, 50-, and 100-image subsets are nested, while the 1,000-image subset is sampled independently. We reproduce the benchmark observation process and use the same 1,000-sample ImageNet validation set as the reference protocol.

Self-generated data setting. When the degradation operator is available, we additionally study a self-generated data setting. At each training epoch, we generate 128 clean samples from the frozen prior and add them to a growing training pool. Their observations are produced on the fly using the known degradation operator. Thus, no real paired training examples are used, although the operator is available during the training. Hyperparameters are selected on a separate set of 25 prior-generated validation samples and their degraded observations.

We use this setting for the known-operator experiments on ImageNet-256, ImageNet-512 and on EARS STFT phase retrieval. We refer to it as the *self-generated* setting.

B.3 Models and training details

Frozen priors. For speech, we train an unconditional NCSN++ model (Song et al., 2021) from scratch on the full clean EARS training set using equation 3 also known as Denoising Score Matching (Song et al., 2021). The model directly predicts the score and uses the standard σ_t^2 loss weighting. All speech representations use complex-STFT magnitude compression $\widetilde{X} = \beta|X|^\alpha e^{j\angle X}$, where $|X|$ and $\angle X$ denote the STFT magnitude and phase, respectively, α controls the magnitude compression, and β is a global scaling factor. Following the complex-STFT speech-restoration SGMSE+ configuration (Richter et al., 2023), we use $\alpha = 0.5$, $\beta = 0.15$, and $\sigma_{\max} = 0.5$ throughout all speech experiments. The resulting prior contains 65.56M parameters and requires 132.70 GFLOPs per NFE for one second of audio.

For images, we use pretrained PixelDiT-XL models (Yu et al., 2026) at 256×256 and 512×512 resolution without further adaptation. We use the epoch-320 checkpoint for ImageNet-256 and the released ImageNet-512 checkpoint. The unconditional velocity field is obtained through the learned null class condition. Architectural configurations and compute per-step for the prior models are reported in Table 3.

LSA models and baselines. As the default architecture for all tasks, the LSA is an NCSN++ model (Song et al., 2021) initialized from scratch. It receives (x, y, t) as input and shares neither parameters nor intermediate features with the frozen prior. For speech, both the prior and LSA are trained in score coordinates. For the main image experiments, the PixelDiT prior is used in its native velocity parameterization and LSA is trained in the velocity parameterization. For an ablation study, we additionally train LSA using a PixelDiT architecture or alternative score- and data-prediction parameterizations (see Appendix A.2). The corresponding results are reported in Tables 8 and 9.

For the PixelDiT baselines, we adapt ControlNet and LoRA to the original frozen PixelDiT backbone. Our ControlNet follows the transformer-based ControlNet-Half implementation of PixArt[4]. A

[3] https://github.com/zhangbingliang2019/DAPS

[4] `https://github.com/PixArt-alpha/PixArt-alpha/blob/master/diffusion/model/nets/pixart_controlnet.py`

Table 3: Prior model configurations used throughout the experiments. All prior parameters remain fixed during downstream training. Speech GFLOPs counts are measured for one second of audio.

Setting	Model	Width	Depth config.	Architecture details	Frozen Parameters	GFLOPs/ NFE
ImageNet-256	PixelDiT-XL (320 ep.)	1152	26	4 pixel blocks; patch size 16	797.38M	311.19
ImageNet-512	PixelDiT-XL	1152	26	4 pixel blocks; patch size 16	797.38M	1352.24
Speech	NCSN++	128	$(1, 1, 2, 2, 2, 2, 2)$	2 residual blocks	65.56M	132.70
	ADM	128	$(1, 1, 2, 2, 2, 2, 2)$	2 residual blocks	72.25M	131.42

trainable branch follows the first half of the patch-level transformer and injects residuals into the corresponding frozen PixelDiT blocks through zero-initialized projections. The observation image is embedded using the PixelDiT patch embedder, while the pixel-level blocks and output head remain unchanged. For our experiments, we use a width-reduced control branch ($d = 692$) while preserving the ControlNet-Half topology. LoRA (Hu et al., 2022) uses the standard low-rank update and is applied to the joint QKV and output projections of all patch- and pixel-level attention blocks. The observation image is encoded by a two-layer conditioning stem and added to the patch tokens, and its final projection is zero-initialized. All original PixelDiT parameters remain frozen.

For the speech baselines, we adapt ControlNet (Zhang et al., 2023) to the frozen NCSN++ score prior. The trainable ControlNet branch follows the encoder and middle block of NCSN++ and injects its features into the corresponding frozen backbone features through zero-initialized 1×1 convolutions. The observation Y is represented by its real and imaginary STFT components and provided to the trainable branch through a zero-initialized input projection. Consequently, the adapted model initially reproduces the frozen prior and gradually learns the observation-dependent correction during conditional score matching. For computational efficiency, we use a reduced-width ControlNet branch with $n_f = 64$, compared with $n_f = 128$ for the frozen prior.

For the DEFT (Denker et al., 2024) baselines we use the authors'[5] implementation and is train on our 128-pair DAPS subsets with the same image augmentations as the other paired methods. Unlike our main LSA setting, the DEFT pipeline uses prior-model outputs and has an optional access to the degradation operator. For speech, we also train a direct conditional posterior model with the same NCSN++ backbone on the corresponding 1- and 10-minute paired subsets as well as on the full training set.

All ControlNet, LoRA and direct posterior models trained on equation 6 with velocity parameterization.

Optimization and augmentation. All LSA models are trained using Adam with a learning rate of 2×10^{-4} and batch size of 64. We use a step learning rate scheduler, and reduce the learning rate to 2.5×10^{-5} through 150 epoches. We use the final training checkpoint and do not select checkpoints based on a metric. To keep the effective number of optimization examples comparable across low-data regimes, small paired datasets are repeated within each training epoch; the underlying clean–observation pairs themselves remain fixed. Image models use the same pair-preserving dihedral augmentation for all methods, consisting of random rotations by multiples of 90° and horizontal flipping. No additional observations are generated from the forward operator in the real-pair low-data setting.

Complexity reporting. All GFLOP counts in Tables 3–5 are measured with `thop`[6] and report the cost of one network function evaluation per image / 1 second of speech. The speech models accept variable-duration inputs.

[5] https://github.com/alexdenker/DEFT

[6] https://github.com/ultralytics/thop

Table 4: Model configurations used across all ImageNet experiments. Parameters and GFLOPs/NFE refer to the trainable LSA model or adapter.

Setting	Model	Width	Depth config.	Architecture details	Trainable params	GFLOPs/ NFE
IN256, 0 real pairs	LSA (NCSN++)	40	$(1, 1, 1, 2)$	2 residual blocks	2.16M	44.30
IN256, 128 pairs	LSA (NCSN++)	32	$(1, 1, 1, 2)$	1 residual block	1.01M	19.84
IN256, 128 pairs	ControlNet	692	13	13 control blocks	124.92M	46.80
IN256, 128 pairs	LoRA	–	30	rank=8	2.15M	1.10
IN256 ablation	LSA (PixelDiT)	96	4	2 pixel blocks; patch size 4	0.98M	44.71
IN512, 0 real pairs	LSA (NCSN++)	40	$(1, 1, 1, 2)$	2 residual blocks	2.16M	176.04

Table 5: Speech model configurations. Parameters and GFLOPs/NFE refer to the trainable model shown in each row. Speech compute is measured for one second of audio.

Setting	Model	Width	Depth config.	Architecture details	Trainable params	GFLOPs/ NFE
1 min	Conditional / LSA	32	$(1, 2, 2)$	2 residual blocks	1.71M	27.60
10 min	Conditional / LSA	64	$(1, 1, 2, 2)$	1 residual block	5.47M	46.51
Full	Conditional / LSA	128	$(1, 1, 2, 2, 2, 2, 2)$	2 residual blocks	65.59M	266.03
0 real pairs Phase Retrieval	LSA	64	$(1, 1, 2, 2)$	2 residual blocks	7.60M	66.22
1 / 10 min	ControlNet	32	7/7	–	1.81M	5.55
Full	ControlNet	64	7/7	–	6.66M	19.64

B.4 Inference refinements

LSA guidance. LSA is trained using the additive posterior-field estimator from equation 8, corresponding to unit LSA weight. At inference, we introduce a time-dependent LSA weight ζ_t,

$$\widehat{f}_\zeta(x, y, t) = f_\theta(x, t) + \zeta_t \widehat{\ell}^f_\psi(x, y, t). \tag{44}$$

We parameterize the guidance schedule by its values at the data and noise endpoints, ζ_0 and ζ_1, and linearly interpolate between them,

$$\zeta_t = (1 - t)\zeta_0 + t\zeta_1. \tag{45}$$

Thus, $\zeta_0 = \zeta_1 = 1$ recovers the field used during training.

LSA scaling serves the same purpose as standard diffusion guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2021): it provides an inference-time control over the strength of conditioning. However, by equation 10, LSA contains both the likelihood score and the prior model error Δ^f_θ. Scaling it therefore does not correspond purely to likelihood tempering, and we interpret ζ_t more generally as a guidance-like control of the learned posterior field.

Restart refinement. We additionally use a short restart-style refinement procedure motivated by restart sampling and posterior restart methods (Xu et al., 2023; Zhang et al., 2025a; Ahmed & Makin, 2026). After completing a reverse trajectory and obtaining an estimate $\widehat{X}_0$, we perturb it to an intermediate time τ of the same Gaussian path,

$$X_\tau = \alpha_\tau \widehat{X}_0 + \sigma_\tau \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{46}$$

and integrate the conditional trajectory again from τ toward the data endpoint using the original observation y. The procedure can be repeated for multiple refinement rounds. The jump time and number of rounds are selected on held-out validation data, and all additional field evaluations are included in the reported NFE. The impact of both inference refinements is ablated in Table 10.

Hyperparameter optimization. Inference hyperparameters are optimized on held-out validation data using Ray Tune with the HyperOpt search algorithm.[7] The search jointly considers the number of integration steps, the endpoint LSA weights (ζ_0, ζ_1), and the restart-refinement parameters when enabled. In our current search space, NFE is varied over integer values from 5 to 30, while ζ_0 and ζ_1 are selected from $[0.0, 5.0]$ in increments of 0.25. The restart time τ is searched within a range of $[0.2, 0.8]$ and by default we use a single restart.

For stochastic sampling and score-parameterized models, we additionally optimize the sampling start time $t_{\text{start}} \in [0.8, 0.999]$ to avoid the endpoint singularity at $t = 1$. We also select between initialization from $\mathcal{N}(0, \sigma^2_{t_{\text{start}}} I)$ and an observation-centered Gaussian $\mathcal{N}(\alpha_{t_{\text{start}}} y, \sigma^2_{t_{\text{start}}} I)$.

We run 150 HyperOpt trials for each task and method. The five best configurations found by the search are subsequently reevaluated on the full validation set, from which the final configuration is selected. The selected configuration is then kept fixed for test evaluation.

For speech enhancement and STFT phase retrieval optimization maximizes PESQ, whereas for bandwidth extension it minimizes LSD. For ImageNet experiments, optimization minimizes LPIPS on 25 held-out training images; in the self-generated data setting, these are replaced by 25 independently generated prior samples and their corresponding degraded observations. The same validation-based optimization procedure is applied to LSA, ControlNet, LoRA, and the direct posterior baseline over the sampling hyperparameters applicable to each method.

C IMAGENET-256 RESULTS

C.1 LOW-DATA IMAGENET-256 RESULTS

Table 6: Detailed ImageNet-256 restoration results using 128 real pairs. We report NFE, PSNR, SSIM, LPIPS, and FID across all evaluated tasks. Bold denotes the best result for each task.

Task	Method	NFE	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Super-resolution (4×)	DEFT	100	24.87	.651	.244	97.15
	ControlNet	43	24.78	.665	.247	89.14
	LSA (ours)	21	**24.93**	**.683**	**.229**	**73.01**
Inpainting (box)	DEFT	100	19.20	**.759**	.194	141.19
	ControlNet	10	**20.24**	.685	.275	152.02
	LSA (ours)	22	18.94	.755	**.193**	**134.00**
Inpainting (random)	DEFT	100	27.00	.805	.156	41.70
	ControlNet	20	23.48	.632	.290	112.99
	LSA (ours)	24	**27.28**	**.808**	**.127**	**32.77**
Gaussian deblurring	ControlNet	42	**24.71**	.650	.262	106.45
	LoRA	36	24.28	.642	.259	106.18
	LSA (ours)	36	**24.86**	**.661**	**.221**	**73.15**
Nonlinear deblurring	ControlNet	55	23.35	.621	.278	119.78
	LoRA	20	23.93	.657	.255	100.87
	LSA (ours)	21	**24.59**	**.731**	**.185**	**52.21**

[7] https://docs.ray.io/en/latest/tune/api/doc/ray.tune.search.hyperopt.HyperOptSearch.html

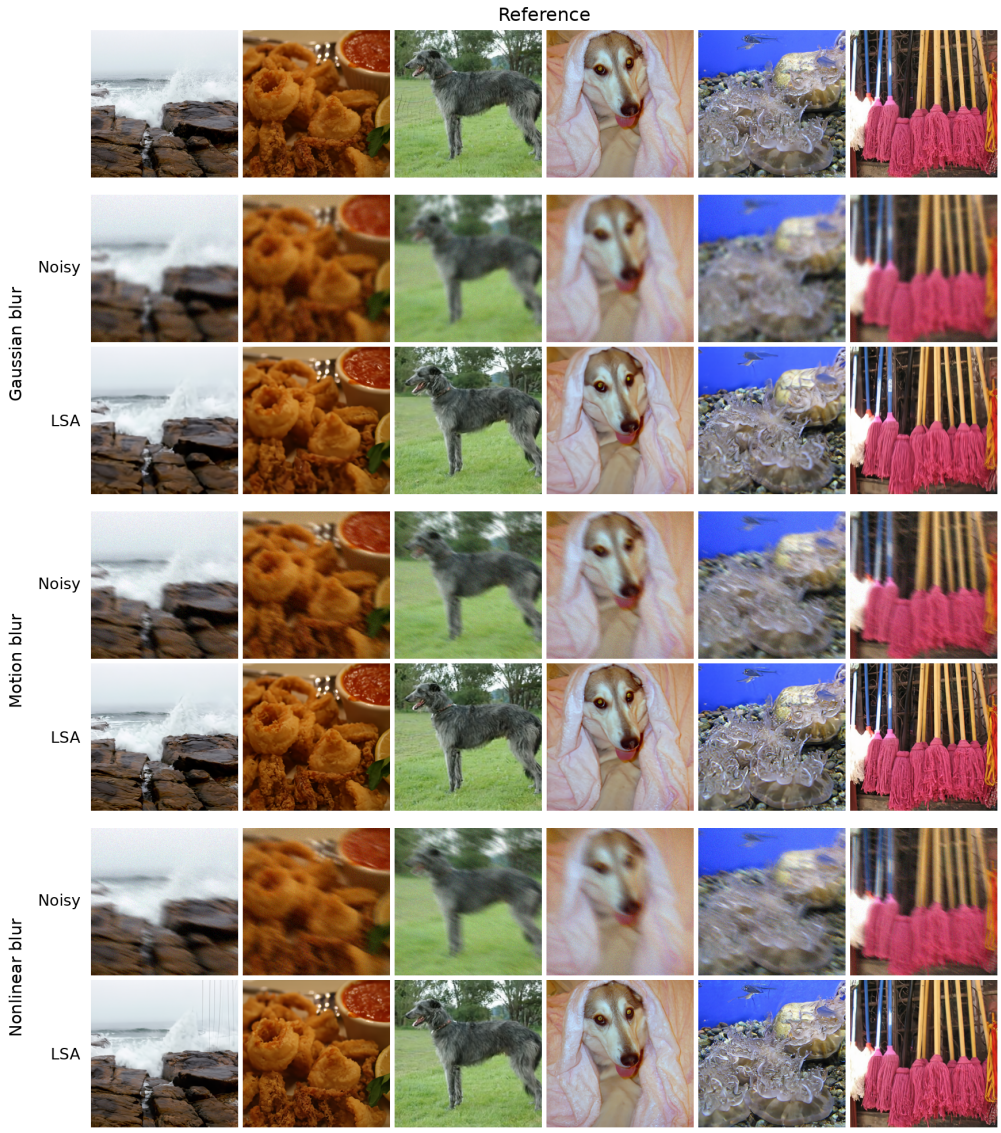

Figure 5: ImageNet-256 blur restoration: references, observations, and LSA (ours) reconstructions.

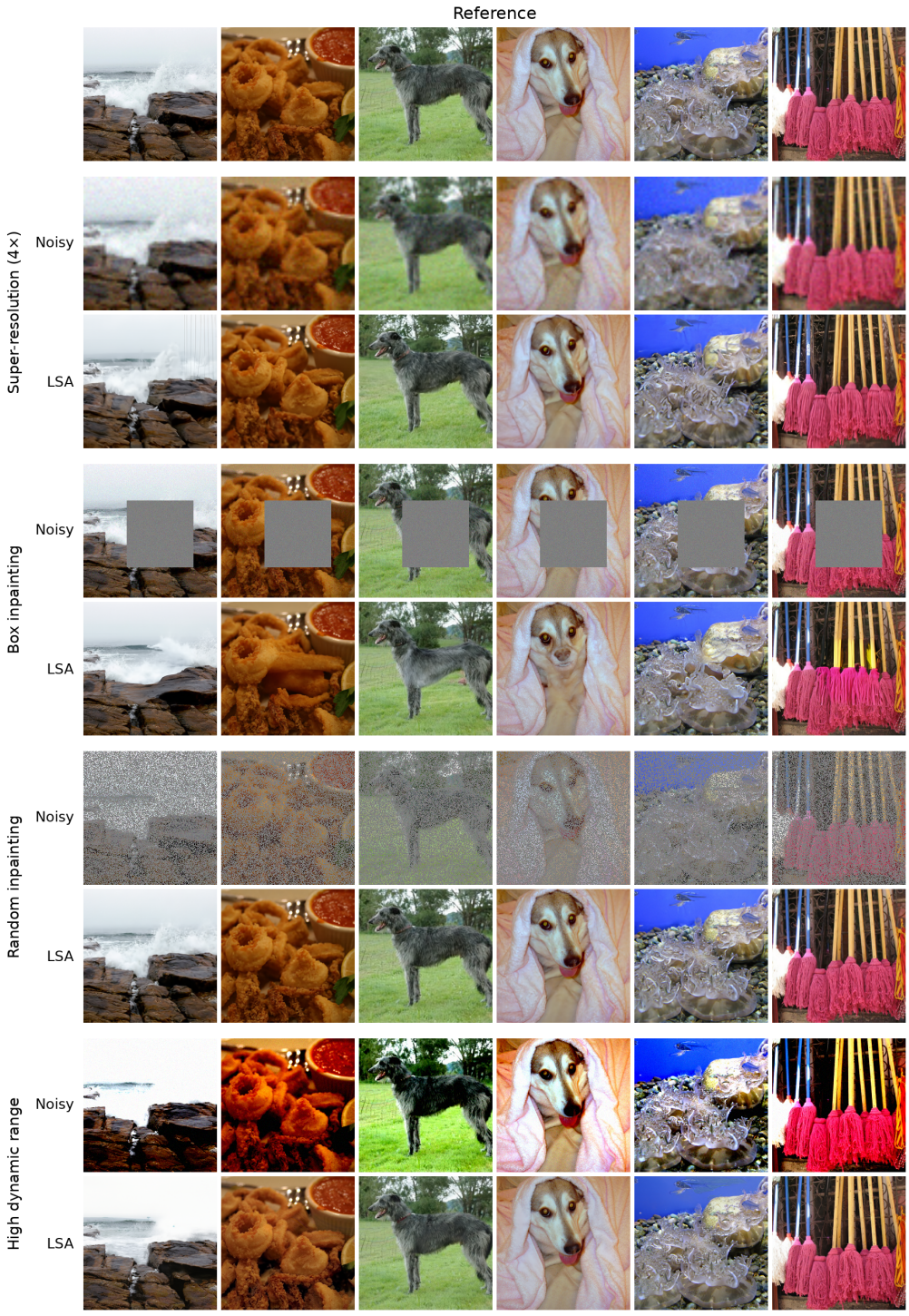

Figure 6: ImageNet-256 super-resolution, inpainting, and HDR restoration results.

D IMAGENET-512 SUPER-RESOLUTION RESULTS

Table 7: ImageNet-512 $8\times$ super-resolution across paired-data budgets. LSA is trained with 25, 50, 100, or 1k real pairs; "self-gen" denotes training on self-generated pairs obtained using the known degradation operator. Baseline results are from Xu et al. (2025). Bold indicates the best available value within each data budget.

Source	Files	NFE ↓	PSNR ↑	LPIPS ↓	FID ↓
DPS+CSE	25	500	22.26	.405	75.64
LSA (ours)	25	12	**23.07**	**.347**	**40.76**
DPS+CSE	50	500	22.84	.353	55.78
LSA (ours)	50	9	**23.20**	**.335**	**39.34**
CSE	100	500	16.54	.484	85.58
DPS+CSE	100	500	22.98	.323	44.59
DMAP+CSE	100	500	**23.58**	.336	39.56
LSA (ours)	100	20	23.32	**.321**	**36.48**
CSE	1k	500	16.68	.466	67.38
DPS+CSE	1k	500	23.35	.314	36.23
DMAP+CSE	1k	500	**23.71**	.325	36.64
LSA (ours)	1k	20	23.26	**.312**	**33.55**
CSE	self-gen	500	17.15	.479	82.32
DPS+CSE	self-gen	500	23.09	.308	38.60
DMAP+CSE	self-gen	500	**23.33**	.311	38.29
LSA (ours)	self-gen	32	23.11	**.297**	**28.90**

E GAUSSIAN DEBLURRING ABLATION

Tables 8 and 9 report the parameterization and architecture ablations, respectively, while Figure 7 shows the corresponding training dynamics.

Table 8: LSA parameterization ablation for Gaussian deblurring on ImageNet-256 with 128 pairs and an NCSN++ backbone. For each parameterization, we report the validation-selected sampling start time t_{start} and initialization. For velocity and data prediction, we additionally report default sampling from the Gaussian endpoint at $t_{\text{start}} = 1$; this setting is not defined for the score parameterization due to the endpoint singularity.

Parameterization	Initialization	t_{start}	NFE	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
Score	Observation marginal	.878	41	24.91	.663	.217	72.86
Velocity	Gaussian	.907	49	24.68	.657	.216	70.23
Velocity	Gaussian	1.000	36	24.86	.661	.221	73.15
Data prediction	Observation marginal	.819	31	25.10	.676	.229	75.72
Data prediction	Gaussian	1.000	29	24.49	.655	.233	74.19

Table 9: LSA backbone ablation for Gaussian deblurring on ImageNet-256 with 128 pairs. We compare NCSN++ and PixelDiT using Gaussian initialization, reporting both the validation-selected sampling start time t_{start} and sampling from the Gaussian endpoint at $t_{\text{start}} = 1$. (Appendix B.4)

FM backbone	Initialization	t_{start}	NFE	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
NCSN++	Gaussian	.907	49	24.68	.657	.216	70.23
NCSN++	Gaussian	1.000	36	24.86	.661	.221	73.15
PixelDiT	Gaussian	.840	22	25.25	.682	.213	68.31
PixelDiT	Gaussian	1.000	19	24.97	.673	.217	71.52

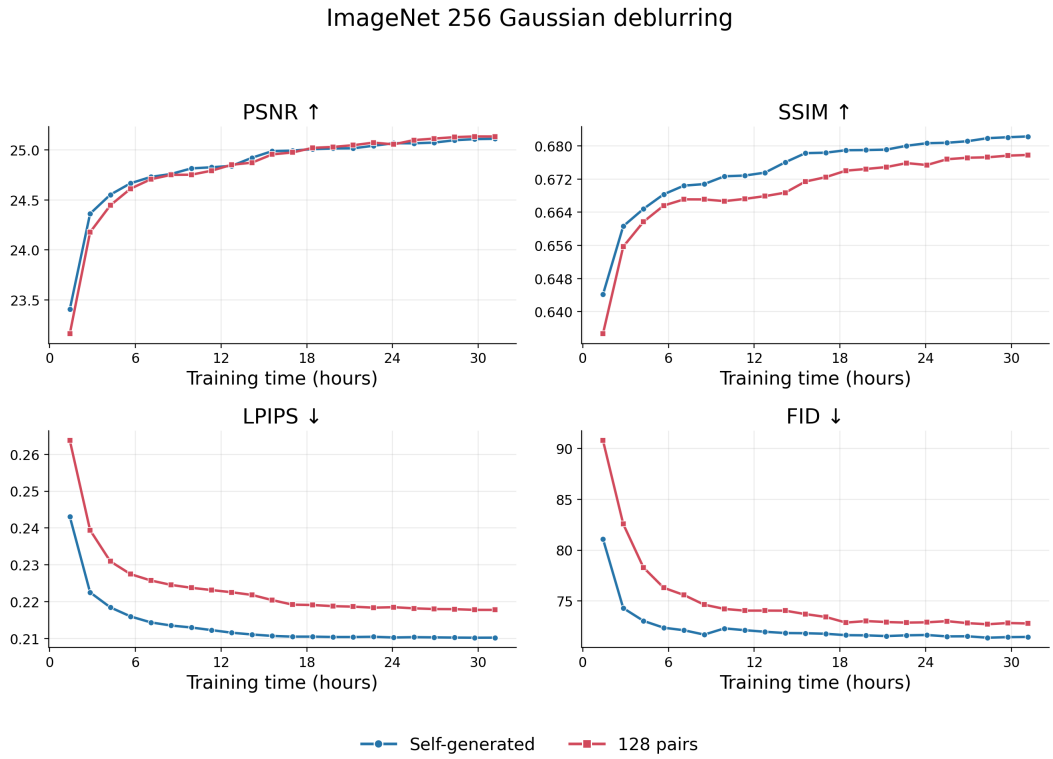

Figure 7: Gaussian-deblurring learning curves for LSA on ImageNet-256 with a PixelDiT prior, trained using either 128 real pairs or self-generated data. Training time is measured on a single A100 GPU.

F BANDWIDTH EXTENSION QUALITATIVE RESULTS

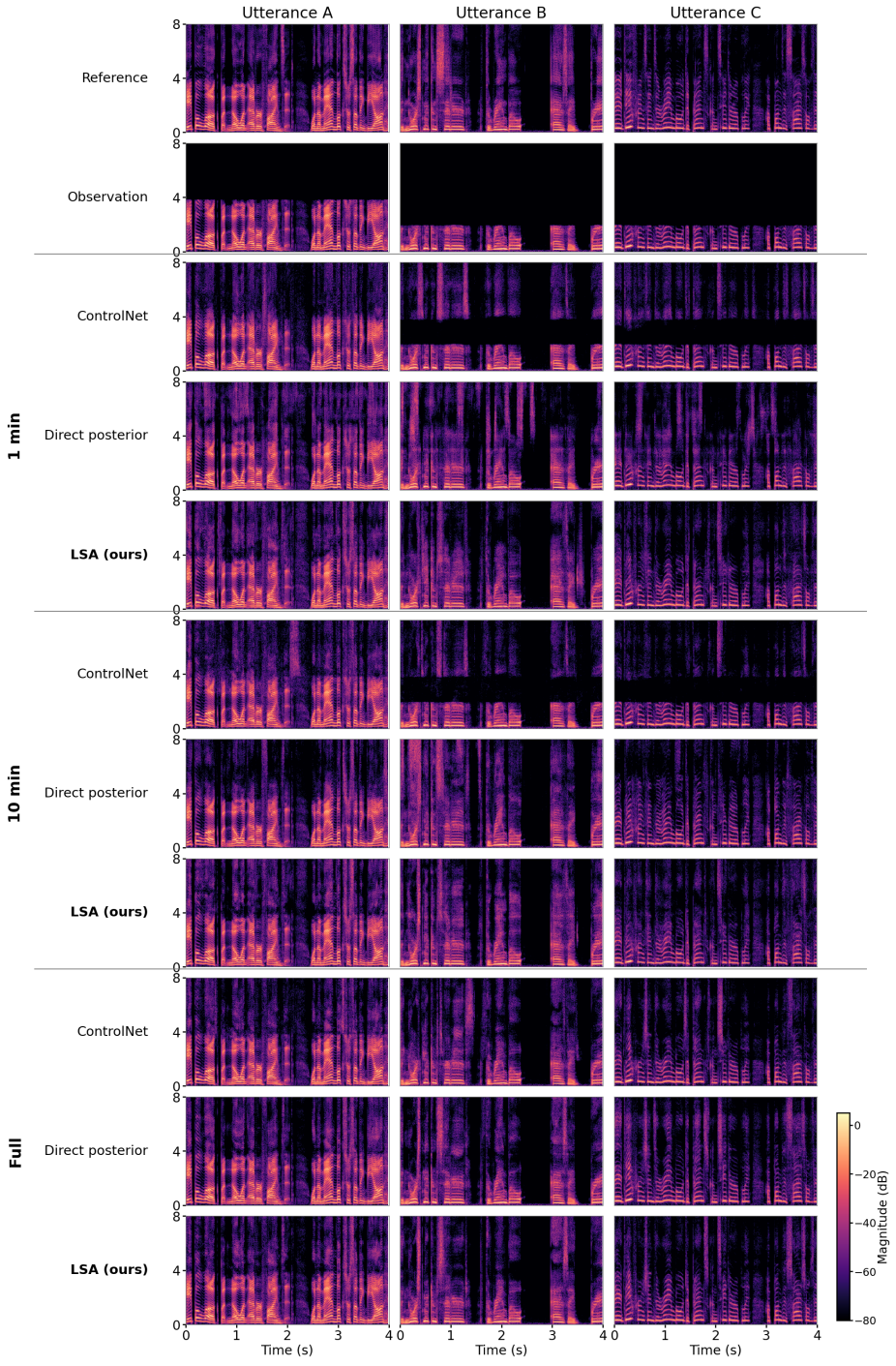

Figure 8: Qualitative BWE comparison on three EARS test utterances. The shared top rows show the clean reference and low-pass-filtered observation; each subsequent block compares ControlNet, direct posterior training, and LSA (ours) using 1 minute, 10 minutes, or the full paired training set. Spectrogram magnitudes are measured relative to the clean-reference peak of each utterance, with frequency shown in kHz.

G PRIOR-SWAPPING ABLATION

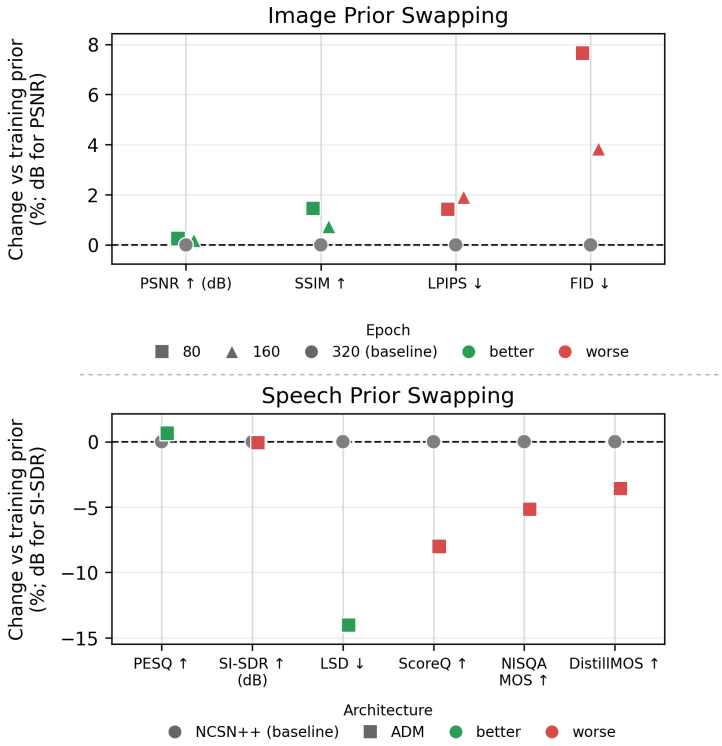

Figure 9: Inference-time prior swapping without retraining LSA. Top: ImageNet-256 Gaussian deblurring with 128 pairs, where the PixelDiT prior used during training (epoch 320) is replaced by earlier checkpoints. Bottom: 1-min speech-enhancement LSA, where the NCSN++ training prior is replaced by an ADM prior trained on the same EARS data. Values show metric changes relative to the training prior; despite the architecture change, performance remains broadly comparable, with a substantial improvement in LSD.

H STOCHASTIC-SAMPLING ABLATION

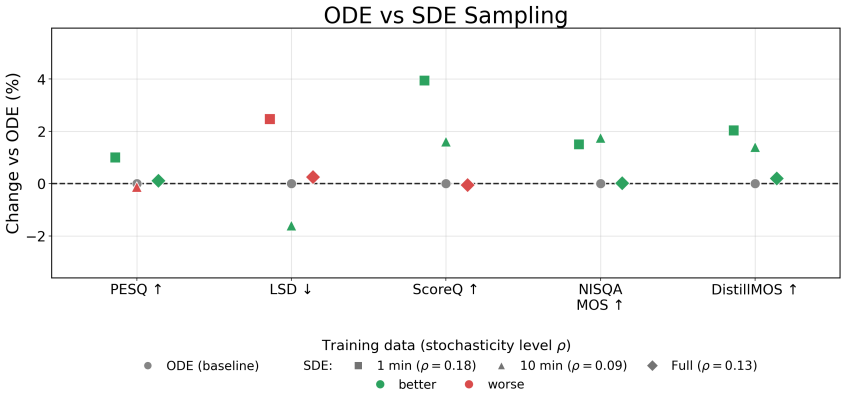

Figure 10: ODE versus SDE sampling for LSA bandwidth extension with 1 min, 10 min, and full paired data. Changes are relative to the ODE baseline ($\rho = 0$); SDE uses $\rho = 0.18$, 0.09, and 0.13, respectively. Mild stochasticity generally improves performance, providing complementary gains.

I INFERENCE REFINEMENT ABLATION

Table 10: Inference-refinement ablation for the 1-min LSA speech-enhancement model. We vary the use of restart refinement and the LSA guidance-weight schedule $\zeta_1 \to \zeta_0$, while keeping the trained model fixed (see Appendix B.4).

Restart	$\zeta_1 \to \zeta_0$	PESQ ↑	SI-SDR ↑	Avg. nMOS ↑	FAD ↓
Yes	$3 \to 0$	1.51	7.82	**3.81**	**0.35**
No	$3 \to 0$	**1.52**	7.67	3.47	1.29
Yes	1	1.47	7.43	3.14	0.70
Yes	2	1.51	**10.56**	2.93	1.82
No	1	1.36	5.96	2.76	1.11