VIPaint: Image Inpainting with Pre-Trained Diffusion Models via Variational Inference

Sakshi Agarwal
Accenture

Gabriel Hope
Swarthmore College

Jimin Heo
Univ. California, Irvine

Erik B. Sudderth
Univ. California, Irvine

Abstract

Diffusion probabilistic models learn to remove noise added during training, generating novel data (e.g., images) from Gaussian noise through sequential denoising. However, conditioning the generative process on corrupted or masked images is challenging. While various methods have been proposed for inpainting masked images with diffusion priors, they often fail to produce samples from the true conditional distribution, especially for large masked regions. Many baselines also cannot be applied to latent diffusion models which generate high-quality images with much lower computational cost. We propose a hierarchical variational inference algorithm that optimizes a non-Gaussian Markov approximation of the true diffusion posterior. Our *VIPaint* method outperforms existing approaches to inpainting, producing diverse high-quality imputations even for state-of-the-art text-conditioned latent diffusion models, and is also effective for other inverse problems like deblurring and superresolution.

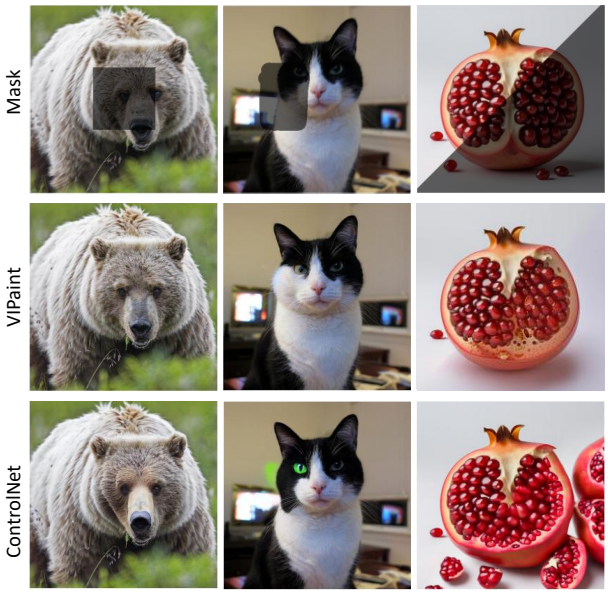

Figure 1: Image inpainting by applying our VIPaint method to pre-trained Stable Diffusion 3.5, and a model retrained via ControlNet (Zhang et al., 2023b).

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2021b; Nichol and Dhariwal, 2021; Song and Ermon, 2019) learn to generate synthetic data by sequentially reducing Gaussian noise across hundreds or thousands of steps, producing deep generative models that have advanced the state-of-the-art in natural image generation (Dhariwal and Nichol, 2021; Kingma et al., 2021a; Karras et al., 2022). Diffusion models for high-dimensional data like images are computationally intensive. Efficiency may be improved by leveraging an autoencoder (Kingma and Welling, 2019; Rombach et al., 2022; Vahdat et al., 2021) to map data to a lower-dimensional space, and then training a diffusion model for the lower-dimensional codes. Such *latent diffusion models* (LDMs) enable efficient but expressive models for images with millions of pixels.

Motivated by the foundational information captured by diffusion models, numerous algorithms have used a pre-trained diffusion model as a prior for tasks such as image editing (Meng et al., 2022), inpainting (Song et al., 2021b; Wang et al., 2023; Kawar et al., 2022; Chung et al., 2022a; Lugmayr et al., 2022; Cardoso et al., 2024; Feng et al., 2023; Trippe et al., 2023; Dou and Song, 2024), and other inverse problems (Kadkhodaie and Simoncelli, 2021; Song et al., 2023; Graikos et al., 2022; Mardani et al., 2024; Chung et al., 2023). Many of these methods are specialized to inpainting with pixel-based diffusion models, a simpler task where every pixel is either perfectly observed or completely

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

missing, and are not easily adapted to state-of-the-art LDMs. Moreover, they are often tested on relatively *easy* restoration tasks where much of the image's global structure is already known. Methods like DPS (Chung et al., 2023) and REDdiff (Mardani et al., 2024) are most effective at inpainting small masked regions, or for tasks like deblurring and super-resolution that only require the refinement of local image details.

RePaint (Lugmayr et al., 2022) and CoPaint (Zhang et al., 2023a) observe that existing methods often produce inconsistent or unrealistic inpaintings in large, contiguous masked regions. Most methods employ an iterative refinement procedure, like that used to generate unconditional samples, and guide their predictions towards the partially observed image via various approximations and heuristics. We hypothesize that this sequential denoising process, from independent Gaussian noise to noise-free images, tends to misrepresent the global structure early in the reverse diffusion trajectory. As the process lacks a mechanism to correct these early errors, the final inpainting can remain globally incoherent when inpainting large regions.

Recent work proposing similar algorithms for image editing (Avrahami et al., 2022) or inpainting (Rout et al., 2023; Corneanu et al., 2024; Chung et al., 2023; Song et al., 2024; Zhang et al., 2025) with LDMs suffers from similar inaccuracies (see Sec. 4). Liu et al. (2024) adapt probabilistic circuits (Choi et al., 2020) for large-mask image inpainting, but their supervised approach must be trained to match a known image mask distribution. Methods based on generative adversarial networks (Zhao et al., 2021; Suvorov et al., 2022) do not leverage foundation models and have similar restrictions, requiring specialized training and many examples of similarly corrupted images for each task. Wang et al. (2024) assume additional side-information, such as segmentations or depths or poses, is available to inpaint large mask regions.

On the other hand, *variational inference* (VI) (Wainwright and Jordan, 2008; Blei et al., 2017) has achieved excellent image restoration results with a wide range of priors, including mixtures (Fergus et al., 2006; Ji et al., 2017) and hierarchical VAEs (Agarwal et al., 2023), but there is little work exploring its integration with state-of-the-art LDMs. While *REDdiff* (Mardani et al., 2024) applies VI to approximate the posterior of pixel-based DMs, its local approximation of the noise-free image posterior is difficult to optimize, requiring annealing heuristics that are sensitive to local optima. MGPS (Moufad et al., 2025) uses VI locally within a sequential denoising procedure to approximately sample from the posterior at a midpoint between the current noise level and the noise-free image.

In this work, we propose *VIPaint*, a novel application of VI that employs both LDMs and pixel-based DMs as priors to handle challenging inference problems, such as large mask image inpainting. VIPaint strategically defines a hierarchical, Markovian and non-Gaussian approximation to the true (L)DM posterior that accounts for a subset of latent noise levels, enabling the inference of both high-level semantics and low-level details from observed pixels *simultaneously*. We efficiently infer variational parameters for each inpainting query, avoiding the need to collect a training set of corrupted images (Liu et al., 2024; Corneanu et al., 2024), expensively fine-tune generative models (Avrahami et al., 2022) or normalizing flow posteriors (Feng et al., 2023) for each query, or retrain large-scale conditional diffusion models (Rombach et al., 2022; Saharia et al., 2022; Nichol et al., 2022; Chung et al., 2022b).

A primary design goal for VIPaint is to avoid the training of a specialized model for each image restoration task, and to instead apply pretrained DMs in a "zero shot" fashion, via variational inference. Our experimental comparisons are thus primarily to the rich literature of methods that adapt pretrained DMs in other ways, but we find that VIPaint may nevertheless outperform specialized inpainting algorithms; see Fig. 1.

We begin by reviewing properties of (latent) diffusion models and prior work on inference with pre-trained diffusion models in Sec. 2. Sec. 3 then develops the VIPaint algorithm, which first fits a hierarchical posterior that best aligns with the observations, and then samples diverse reconstructions from this posterior. Results in Sec. 4 on inpainting, as well as linear deblurring and superresolution, show substantial qualitative and quantitative improvements in producing inpaintings that capture the richness of contemporary DMs.

2 BACKGROUND

2.1 Denoising Diffusion Generative Models

The diffusion process begins with clean data x, and defines a sequence of increasingly noisy representations of x. We denote these *latent variables* by $z_t \in \{z_0, ..., z_T\}$, where the *time* t runs from $t = 0$ (low noise) to $t = T$ (substantial noise). The distribution of z_t given x, for any time $t \in [0, T]$, is

$$q(z_t \mid \bar{x} = \texttt{enc}(x)) = \mathcal{N}(z_t \mid \alpha_t \bar{x}, \sigma_t^2 I), \qquad (1)$$

where $\sigma_t > 0$ and strictly increases with t. For LDMs, $\bar{x} = \texttt{enc}(x)$ where $\texttt{enc}(\cdot)$ is a pre-trained encoder function that maps x to a lower-dimensional latent code for sampling efficiency. For pixel-based DMs, $\texttt{enc}(x) = x$.

Diffusion induces a Markov chain for which the conditionals $q(z_t \mid z_s)$, $q(z_s \mid z_t, \bar{x})$ are tractable Gaussians for any $s < t$ (see App. B). The signal-to-

noise ratio (Kingma et al., 2021b) induced by this diffusion process at time t equals $\mathrm{SNR}(t) = \alpha_t^2/\sigma_t^2$. The SNR monotonically decreases with time, so that $\mathrm{SNR}(t) < \mathrm{SNR}(s)$ for $t > s$. This DM specification includes variance-preserving diffusions (Ho et al., 2020; Sohl-Dickstein et al., 2015) where $\alpha_t = \sqrt{1-\sigma_t^2}$, variance-exploding diffusions (Song and Ermon, 2019; Song et al., 2021b) where $\alpha_t = 1$, and rectified flow models (Liu et al., 2022) where $\alpha_t = 1 - \sigma_t$.

Image Generation. We generate novel images by reversing the diffusion process of Eq. (1), inducing a hierarchical generative model that samples a sequence of latent variables z_t before sampling x. Generation progresses backward in time from $t = T$ to $t = 0$ via a finite temporal discretization into $T \approx 1000$ steps, either uniformly spaced as in discrete diffusion models (Ho et al., 2020), or via a possibly non-uniform discretization (Karras et al., 2022) of an underlying continuous-time stochastic differential equation (Song et al., 2021b). Letting $t-1$ be the timestep preceding t, the generative model for data x is expressed as:

$$p_\theta(x) = \int_z p(z_T) p(x \mid z_0) \prod_{t=1}^{T} p_\theta(z_{t-1} \mid z_t)\, dz. \tag{2}$$

The marginal distribution of z_T is typically a spherical Gaussian $p(z_T) = \mathcal{N}(z_T \mid 0, \sigma_T^2 I)$. Pixel-based diffusion models take $p(x \mid z_0)$ to be a simple factorized likelihood (Kingma et al., 2021a) for each pixel in x, $p(x \mid z_0) \propto q(z_0 \mid x)$, while LDMs define $p(x \mid z_0)$ via a pre-trained decoder neural network so that $\mathbb{E}[x \mid z_0] = \mathtt{dec}(z_0)$. The conditional latent distribution $p_\theta(z_{t-1} \mid z_t)$ maintains the Gaussian form $q(z_{t-1} \mid z_t, \bar{x})$ induced by the forward noise process,

$$\begin{aligned} p_\theta(z_{t-1} \mid z_t) &= q(z_{t-1} \mid z_t, \bar{x} = \hat{x}_\theta(z_t, t)) \\ \text{where} \quad \hat{x}_\theta(z_t, t) &= \frac{z_t - \sigma_t \hat{\epsilon}_\theta(z_t, t)}{\alpha_t}, \end{aligned} \tag{3}$$

but with the encoded data $\bar{x} = \mathtt{enc}(x)$ approximated via a differentiable noise predictor $\hat{\epsilon}_\theta(z_t, t)$. The denoising neural network may incorporate U-Net (Ronneberger et al., 2015) or transformer (Peebles and Xie, 2023) architectures, and is trained to optimize a variational lower bound (Ho et al., 2020; Song et al., 2021b) of the marginal likelihood of data x,

$$\begin{aligned} -\log p_\theta(x) \le \mathcal{L}(\theta; x) = C + \\ \frac{T}{2} \mathbb{E}_{t,\epsilon,x} \left[\left(\frac{\sigma_t^2 \alpha_{t-1}^2}{\alpha_t^2 \sigma_{t-1}^2} - 1 \right) \|\epsilon - \hat{\epsilon}_\theta(\alpha_t \bar{x} + \sigma_t \epsilon, t)\|_2^2 \right], \end{aligned} \tag{4}$$

for a constant C. The expectation is over $x \sim p_{\mathrm{data}}(x)$, $\epsilon \sim \mathcal{N}(0, I)$, $t \sim \mathrm{Uniform}(1, T)$. Some DMs drop the SNR weights in (4) during training (Ho et al., 2020).

Inference with Diffusion Models. Image inpainting (or more broadly, data imputation) tasks arise when we are given partial observations $y = x \odot m$, where m is a binary mask indicating missing pixels. Recovering x from y is challenging, especially when large image regions are masked, because many x could produce the same observation y. Inpainting is an example of a broader class of *linear inverse problems* which also includes tasks such as deblurring and super-resolution. To express the posterior $p_\theta(x \mid y)$ given a DM prior, we adapt the generative process of Eq. (2):

$$p_\theta(x|y) = \int_z p_\theta(z_T|y) p_\theta(x|z_0, y) \prod_{t=1}^{T} p_\theta(z_{t-1}|z_t, y)\, dz. \tag{5}$$

Exactly evaluating this predictive distribution is infeasible due to the non-linear noise prediction and decoder networks, which make $p_\theta(z_{t-1} \mid z_t, y)$ intractable. Several heuristic methods for approximately sampling from z given y are discussed in App. B. We instead develop a variational inference algorithm that more accurately approximates $p_\theta(x, z \mid y)$.

2.2 Variational Inference of Missing Data

REDdiff (Mardani et al., 2024) uses pixel-based DMs as priors and defines a Gaussian variational distribution $p_\theta(x \mid y) \approx q_\lambda(x) = \mathcal{N}(x \mid \mu, \sigma^2 I)$ over the data space, where $\lambda = \mu$ and the variance is fixed to the same small constant $\sigma^2 \approx 0$ for all pixels. By reducing the posterior approximation to a *single* image μ, REDdiff induces a simple variational inference objective:

$$D\big(q_\lambda(x) \,\|\, p(x|y)\big) = -\log p(y|\mu) + D\big(q_\lambda(x) \,\|\, p_\theta(x)\big) \tag{6}$$

REDdiff seeks an image μ that reconstructs the observation y (at the pixels not occluded by the mask m), while simultaneously having high probability (low KL divergence) under the prior. This diffusion regularizer decomposes as an expectation over many times.

While the loss of Eq. (6) is simple, Mardani et al. (2024) find direct optimization to be difficult and unstable, and find that annealing time from $t = T$ to $t = 0$ (as in standard diffusion samplers) outperforms unbiased optimization of the variational bound through random time sampling. Visual examples in the Appendix compare REDdiff-V, which uses random-time sampling as justified by the correct variational bound, and REDdiff which gradually anneals time from T to 0. REDdiff also does not propagate gradients through the denoising network $\epsilon_\theta(z_t, t)$, as optimization of the true variational bound would require, to prevent optimization instability. We hypothesize that this instability is due to the denoising function's lack of smoothness at low noise levels (Yang et al., 2024).

Because REDdiff employs a simple variational posterior that directly optimizes an image at the noise-free ($t = 0$) level only, it is inherently incapable of capturing uncertainty in x, and instead seeks a single posterior mode. Additionally, its optimization process is

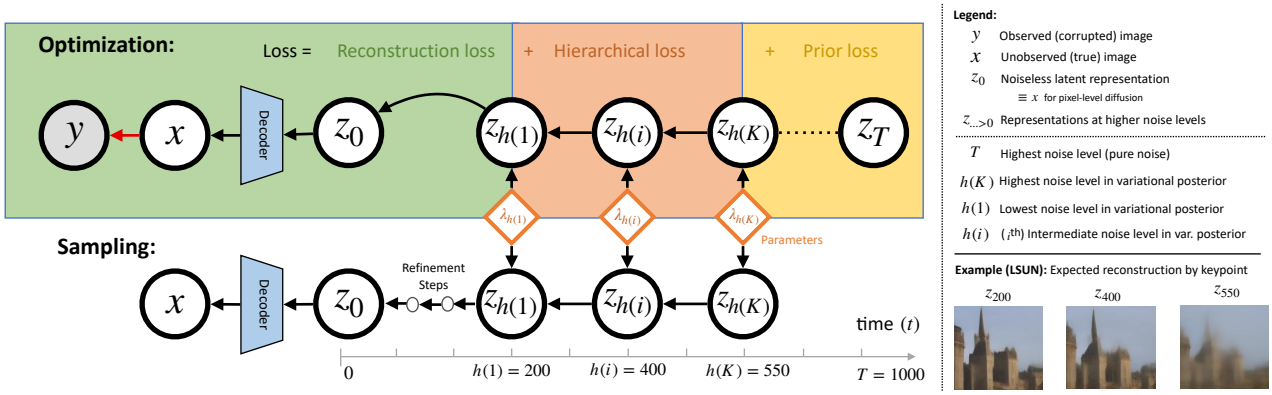

Figure 2: The hierarchical approximate posterior of VIPaint is defined over a coarse sequence of intermediate latent steps, or keypoints, between $h(K)$ and $h(1)$. During optimization, the variational parameters λ defining the posterior at these sparse times are fit via a prior loss on times above $h(K)$, a hierarchical loss defined across K keypoints, and a reconstruction loss estimated using a sample-based one-step approximation of $p_\theta(x \mid z_{h(1)})$. After a single variational optimization, multiple samples may be drawn via gradient-based stochastic refinement.

biased because it relies on annealing time during the diffusion process rather than randomly sampling time points. We demonstrate that our VIPaint framework better models posterior uncertainty, enables stable optimization of an unbiased variational bound, and is applicable to both pixel-based and latent DMs.

3 VARIATIONAL INFERENCE OF LATENT DIFFUSION PATHS

Given a pre-trained diffusion model, VIPaint approximates the distribution of latent variables z_t given a test observation y. It defines a $K-$step hierarchical posterior over a coarse sequence of mid-ranged timesteps $\{h(i)\}_{i=1}^{K}$, where $h(1)$ and $h(K)$ mark the start and end timepoints in the hierarchy. This posterior is defined by variational parameters λ optimized, for each image without amortization (Kingma and Welling, 2019), via a variational bound. After optimization, we perform iterative, gradient-based refinements of posterior samples in the low-noise latent space $[0, h(1)]$ to generate final inpaintings. See Fig. 2 for an overview.

VIPaint offers several practical advantages over RED-diff (Mardani et al., 2024), as the latent-space hierarchical posterior: 1) infers global-to-local semantics in the latent space, consistent with the corrupted image y; 2) accounts for uncertainty in missing pixels; 3) strategically avoids training instabilities (Yang et al., 2024) which arise in the low-noise latent space $[0, h(1))$; and 4) easily extends to latent DMs. Below, we detail VIPaint's Markov posterior, optimization, and sampling strategies for diverse inpaintings.

Variational Posterior Formulation. VIPaint defines a latent-space hierarchical posterior over a subset of K *keypoints* that capture the most informative phases of the reverse diffusion process:

$$q_\lambda(z_{h(1):h(K)}) = \left(\prod_{i=1}^{K-1} q_\lambda(z_{h(i)} \mid z_{h(i+1)})\right) q_\lambda(z_{h(K)}).$$

Here, $K \geq 2$ and $h(i)$ is the time of the keypoint preceding $h(i+1)$. Experiments suggest tuning these keypoints to capture intermediate-noise timesteps with SNR (α_t^2/σ_t^2) in the range $[0.2, 0.5]$ across different (latent) DMs, see Sec 4.2. For the highest timestep $h(K)$, we let $q_\lambda(z_{h(K)}) = \mathcal{N}(z_{h(K)} \mid \mu_{h(K)}, \tau_{h(K)})$ be a factorized Gaussian. For lower-noise timesteps,

$$\begin{aligned} &q_\lambda(z_{h(i)} \mid z_{h(i+1)}) \\ &= \mathcal{N}(z_{h(i)} \mid \gamma_{h(i)} \bar{z}_{h(i)} + (1-\gamma_{h(i)})\mu_{h(i)}, \tau^2_{h(i)}), \qquad (7) \\ &\bar{z}_{h(i)} = E[z_{h(i)} \mid z_{h(i+1)}, \bar{x} = \hat{x}_\theta(z_{h(i+1)}, h(i+1))]. \end{aligned}$$

The standard deviation $\tau_{h(i)}$ is learned and varies across dimensions of the data (or latent code for LDMs), allowing the posterior to dynamically increase uncertainty in regions with more masked image pixels. The mean is a convex combination (with learned, dimension-specific weights $\gamma_{h(i)}$) of the prior diffusion prediction $\bar{z}_{h(i)}$ and an image-specific variational parameter $\mu_{h(i)}$. Intuitively, optimizing $\mu_{h(i)}$ allows predictions from the DM prior to be perturbed towards means consistent with the observation y. Note that the overall variational approximation is Markov (like the true posterior) but non-Gaussian, due to the incorporation of the non-linear denoising network $\hat{x}_\theta(\cdot)$.

Some sampling-based inpainting methods (Song et al., 2021b; Lugmayr et al., 2022; Kawar et al., 2022; Song et al., 2024) also linearly combine observations y with samples z_t, but employ either hard constraints or manually-tuned weights. VIPaint instead incorporates free parameters $\lambda = \{\mu_{h(K)}, \tau_{h(K)}, (\gamma_{h(i)}, \mu_{h(i)}, \tau_{h(i)})_{i=1}^{K-1}\}$ across K latent levels, defined over each pixel in the image or its encod-

ing. This flexible posterior is key to effectively reusing the diffusion prior *and* aligning precisely with a particular observation y, without the need to re-train a specialized, conditional DM. We use y to initialize $\mu_{h(i)}$ by scaling its encoding $\texttt{enc}(y)$ by the forward diffusion constant $\alpha_{h(i)}$. We use the DM noise schedule to automatically initialize posterior variances, see App. E.

Fitting the Variational Posterior. We optimize a variational lower bound on the marginal likelihood of the observation y. As derived in App. C,

$$L(\lambda) = \underbrace{-\mathbb{E}[\log p_\theta(y|z_{h(1)})]}_{\text{reconstruction loss}} + \underbrace{D\Big[q_\lambda(z_{h(K)})||p_\theta(z_{h(K)})\Big]}_{\text{prior loss}}$$
$$+ \underbrace{\sum_{i=1}^{K-1} \mathbb{E}_{z_{h(i+1)}} D\Big[q_\lambda(z_{h(i)}|z_{h(i+1)}) \;||\; p_\theta(z_{h(i)}|z_{h(i+1)})\Big]}_{\text{hierarchical loss}}. \tag{8}$$

VIPaint seeks latent-posterior distributions that assign high likelihood to the observed features y (by minimizing the reconstruction loss), while simultaneously aligning with the medium-to-high noise levels encoding image semantics (hierarchical and prior losses). Expectations are with respect to the hierarchical approximate posterior $q_\lambda(z_{h(1):h(K)})$. We approximate $L(\lambda; y)$ with M Monte Carlo samples (typically 5-10) from $q_\lambda(z_{h(1):h(K)})$, via an ancestral sampler that proceeds from high to low noise: $z^{(m)}_{h(K)} \sim q_\lambda(z_{h(K)}), z^{(m)}_{h(i)} \sim q_\lambda(z_{h(i)}|z^{(m)}_{h(i+1)})$ for $i = K-1, \ldots, 1$. We use automatic differentiation, with reparameterized (Kingma and Welling, 2019) representation of samples from q_λ, to compute gradients with respect to λ. This allows end-to-end optimization through multiple applications of the denoising network $\hat{x}_\theta(\cdot)$, with no annealing.

Reconstruction Loss. This term guides the posterior to align its samples $z_{h(1)}$ with observations y, but it is intractable to analytically integrate over both x and z_0. Estimation via sampling is possible, but expensive as it would require backpropagation through multiple sampling steps. We instead adopt the approximation of (Rout et al., 2023; Chung et al., 2023):

$$p(y \mid z_{h(1)}) \approx p(y \mid \texttt{dec}(\hat{x}_\theta(z_{h(1)}, h(1))). \tag{9}$$

For non-latent DMs, $\texttt{dec}(x) = x$. As $h(1)$ is close to $t = 0$, this approximation leads to updates in the posterior parameters accurate enough to guide samples of $z_{h(1)}$ to be consistent with y. As discussed below, after optimization samples of z_0 are drawn conditioned on y and $z_{h(1)}$ using a more fine-grained process that samples *all* intermediate steps between $z_{h(1)}$ and z_0.

We use the L_1 reconstruction loss (Laplace log-likelihood) for $\log p(y \mid x)$, and add a perceptual loss term (Zhang et al., 2018) when using LDMs, to better match the objective originally used to train the decoder and reduce blur (see Appendix).

Prior Loss. As derived in the Appendix,

$$D\Big(q_\lambda(z_{h(K)}) \,\|\, p_\theta(z_{h(K)})\Big) = \tag{10}$$
$$\frac{T - h(K)}{2}\, \mathbb{E}\Big[D\big(q(z_{t-1}|z_t, z_{h(K)}) \,\|\, p_\theta(z_{t-1}|z_t)\big)\Big].$$

The expectation is over t uniformly sampled in $[h(K), T]$ instead of the entire range $[0, T]$, $z_{h(K)} \sim q_\lambda(z_{h(K)})$, and $z_t \sim q(z_t \mid z_{h(K)})$. This loss regularizes the samples $z_{h(K)}$ to follow the high-level image semantics implicitly encoded by diffusions in $[h(K), T]$.

Hierarchical Loss. The hierarchical loss term further regularizes posterior samples $\{z^{(m)}_{h(i)}\}_{i=1}^{K-1}$ to capture high-to-mid-level image details in the critical intermediate range (Karras et al., 2022) of noise levels $[h(1), h(K)]$ in the latent diffusion space. As with the prior loss, the hierarchical loss can be estimated via sampling (see Appendix); given a sample $z^{(m)}_{h(i+1)}$, $D\Big(q_\lambda(z_{h(i)} \mid z^{(m)}_{h(i+1)}) \,||\, p_\theta(z_{h(i)} \mid z^{(m)}_{h(i+1)})\Big)$ can be computed analytically. We simplify computation by aligning the time discretization of the prior to the posterior keypoints, as in methods for accelerating unconditional DM sampling (Song et al., 2021a), and thus avoid the need to sample times between keypoints.

Optimization. The number of optimization steps may be chosen to flexibly trade speed for accuracy. If the posterior is only defined on the noise-free level z_0 as in REDdiff (Mardani et al., 2024), the VIPaint objective of Eq. (8) degenerates to their (non-annealed) variational bound. However, VIPaint strategically avoids low noise levels in its posterior, avoiding the instabilities that substantially reduce REDdiff performance, and enabling generalization to LDMs.

Sampling. After optimization, the hierarchical posterior $q_\lambda(z_{h(1):h(K)})$ is *semantically* aligned with the observation. We employ ancestral sampling on our K-level hierarchical posterior, from $h(K)$ to $h(1)$, to generate samples $z_{h(1)}$ as in Fig. 2. This step gradually adds diverse image details. VIPaint then refines $z_{h(1)}$ using the prior denoising model at every step $t < h(1)$. Similar to DPS (Chung et al., 2023) and the Langevin dynamics (Song and Ermon, 2019) underlying unconditional DM samplers, we update the samples using the gradient of the likelihood $\log p_\theta(y \mid z_t), t < h(1)$, approximated as above. This ensures fine-grained details of our final inpaintings are consistent and realistic. By using a variational posterior to explicitly capture diffusion processes at moderate and high noise levels, and only relying on local gradient-based updates for low noise levels, our VIPaint method substantially improves on direct Langevin samplers like DPS.

Table 1: Quantitative ImageNet64 Inpainting Results.

Method	Rotated Window			Random Mask		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
VIPaint-2	<u>9.24</u>	**0.56**	**0.30**	**13.33**	**0.62**	**0.23**
CoPaint-TT	8.51	0.51	<u>0.32</u>	<u>12.51</u>	0.58	<u>0.25</u>
CoPaint	8.47	0.50	0.35	12.12	0.56	0.28
RePaint	8.82	**0.56**	<u>0.32</u>	12.05	<u>0.59</u>	0.26
DPS	8.15	<u>0.53</u>	<u>0.32</u>	11.45	0.56	0.29
Blended	7.68	0.52	0.34	11.47	0.57	0.28
RedDiff	8.56	0.45	0.46	11.89	0.51	0.41
RedDiff-V	**9.27**	<u>0.53</u>	0.41	8.35	0.16	0.67

Using the pixel-based EDM prior for all methods, PSNR, SSIM, and LPIPS are averaged over 1000 inpaintings. VIPaint shows the best performance (**bold**), and the second best is <u>underlined</u>.

Table 2: Quantitative SD3.5 Inpainting Results.

Method	PSNR↑	SSIM↑	LPIPS↓
VIPaint-2	**26.69**	**0.946**	**0.050**
DPS	25.22	0.943	0.061
ControlNet	22.89	0.703	0.137
SD-Inpaint	23.50	0.692	0.127

Results are averaged across 10 sample completions for 100 synthetic images generated using SD3.5.

4 EXPERIMENTS & RESULTS

We first experiment with three natural image datasets that have been widely used to evaluate image restoration and inpainting with DMs: LSUN-Church (Yu et al., 2015), ImageNet64, and ImageNet256 (Deng et al., 2009). For ImageNet64, we use the pre-trained class-conditioned pixel-space *EDM* diffusion model (Karras et al., 2022). For LSUN-Churches256 and ImageNet256, we use the pre-trained LDMs from Rombach et al. (2022), as is standard in prior work. We sample (*not* cherry-pick) 100 or 1000 test images for each dataset, and match our experimental settings and preprocessing to previous work (DPS, Chung et al. (2023)). We consider three masking patterns: 1) as in the easier experimental setup of REDdiff (Mardani et al., 2024), 1000 images with a small mask (under 30% of image) adapted from Palette (Saharia et al., 2022); 2) 100 images using a random mask (Zhao et al., 2021) covering 40-80% of image; 3) 100 images using a randomly rotated mask covering about 50% of image. Given the larger uncertainty for patterns 2-3, we sample and evaluate 10 reconstructions per test image (1000 reconstructions total). We emphasize that experimental setups that mask large portions of each image provide a far more challenging inpainting benchmark. (Note that several weak-performing baselines only evaluated on the easy, small mask scenario.)

We also demonstrate that VIPaint is applicable to the latest, foundational text-to-image diffusion models via further experiments with *Stable Diffusion* (SD) 3.5 (Esser et al., 2024). To focus quantitative comparisons on inference performance, while avoiding potential biases from the choice of prompt, our quantitative experiments use 100 images generated from the SD 3.5 model and provide the same prompt for inpainting (see examples in Appendix). Some recent methods (Spagnoletti et al., 2025; Kim et al., 2025) refine the textual prompt as part of the inpainting process; incorporating prompt refinement with VIPaint is a promising direction for future research. For qualitative comparison, we also include real images with prompts from the MSCOCO 2014 validation set (Lin et al., 2014). All SD experiments use 1024×1024 images.

We use the notation VIPaint-K to denote the number of keypoints K in the hierarchical VIPaint posterior. We found empirically that discretizations and hyperparameters of VIPaint translate well between models using the same noise schedule, as demonstrated by the LSUN and ImageNet-256 latent diffusion models.

Baselines. We compare VIPaint with several methods designed for pixel-based DMs: *i)* blending methods, *Blended* (Song et al., 2021b) and *RePaint* (Lugmayr et al., 2022); *ii)* sampling methods, *DPS* (Chung et al., 2023) and *CoPaint* (Zhang et al., 2023a); *iii)* the *REDdiff* (Mardani et al., 2024) variational approximation. Although not exhaustive, these methods exemplify recent developments in image inpainting via DMs. For LDMs we compare VIPaint with *DPS*, *PSLD* (Rout et al., 2023), *MGPS* (Moufad et al., 2025), and *ReSample* (Song et al., 2024). We omit other recent methods, such as DAPS (Zhang et al., 2025) and LATINO (Spagnoletti et al., 2025), which reported reduced performance for inpainting compared to previous work. We report the Peak Signal-To-Noise Ratio (PSNR), Structural Similarity (SSIM, Wang et al. (2004)), Kernel Inception Distance (KID, Bińkowski et al. (2018)), and Learned Perceptual Image Patch Similarity (LPIPS, Zhang et al. (2018)) metrics. We show examples of LDM inpaintings in Fig. 3, Fig. 1, and the Appendix.

Related work using Stable Diffusion (SD) for inverse problem inference (Zhang et al., 2025; Rout et al., 2023) has been limited to earlier versions (SD v1.5 and SD v2). We thus compare our SD results to our own re-implementation of DPS, and two methods that fine-tune (with substantial training cost) SD specifically for inpainting: ControlNet (Zhang et al., 2023b) and SD-Inpainting (Esser et al., 2024).

4.1 Image Inpainting Results

VIPaint enforces consistency with large masks. The results in Tables 1 and 3 show that prior methods perform well for small masks, while for large masks

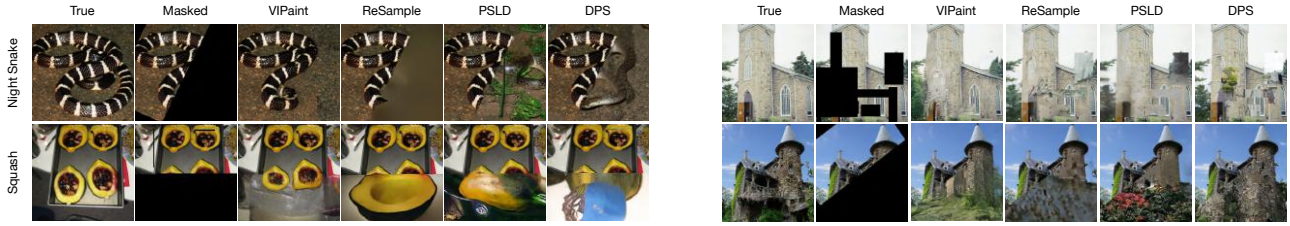

Figure 3: Inpainting with large (random or rotated window) masks using LDMs for Imagenet256 (left) and LSUN churches (right). DPS, PSLD, and ReSample produce blurry inpaintings. Despite being conditioned on class labels, baseline methods' inpaintings for ImageNet are inconsistent with the observed image. In contrast, VIPaint captures global semantics, producing highly realistic inpaintings. See Appendix for more examples.

Table 3: Quantitative 256×256 Latent DM Inpainting Results.

		Small Mask				Rotated Window				Random Mask			
	Method	PSNR↑	SSIM↑	LPIPS↓	KID*↓	PSNR↑	SSIM↑	LPIPS↓	KID*↓	PSNR↑	SSIM↑	LPIPS↓	KID*↓
Imagenet	VIPaint-2	13.51	0.47	0.30	**2.80**	**9.43**	0.44	**0.39**	**5.40**	**10.04**	**0.53**	**0.39**	**6.10**
	MGPS	**20.84**	**0.86**	**0.16**	3.30	7.89	**0.53**	0.40	16.20	9.26	**0.53**	**0.39**	25.60
	ReSample	15.36	0.58	0.37	11.00	7.58	0.40	0.51	26.50	9.57	0.42	0.48	28.70
	PSLD	14.72	0.52	0.41	11.00	7.51	0.33	0.54	18.30	9.50	0.34	0.52	16.80
	DPS	14.62	0.51	0.42	12.00	7.35	0.32	0.50	15.90	9.25	0.33	0.49	20.50
LSUN	VIPaint-2	16.18	0.61	0.20	**0.30**	**8.39**	0.33	0.45	**7.40**	9.58	0.34	0.44	6.60
	MGPS	**19.02**	**0.84**	**0.16**	4.30	7.40	0.33	**0.40**	28.60	**9.88**	**0.56**	**0.36**	24.40
	ReSample	17.21	0.64	0.38	10.90	8.04	**0.43**	0.54	30.90	8.95	0.41	0.56	**6.00**
	PSLD	13.63	0.43	0.53	2.40	7.00	0.34	0.58	8.30	8.33	0.32	0.58	7.20
	DPS	13.17	0.46	0.56	48.00	7.59	0.32	0.61	9.80	8.81	0.31	0.61	7.40

For LDMs of the ImageNet256 and LSUN-Churches256 datasets, the PSNR, SSIM, LPIPS, and KID metrics are the mean score across 1000 inpaintings. For compactness, we report KID* = KID $\times 10^3$.

we see a clear improvement with VIPaint. For pixel-based DMs, both RedDiff and DPS perform poorly. RePaint, CoPaint, and CoPaint-TT show relative improvements, but do not match VIPaint across any dataset or masking pattern. Notably CoPaint-TT integrates the "time travel" heuristic proposed by RePaint (Lugmayr et al., 2022) with CoPaint, requiring more time than both CoPaint and VIPaint-2, but nevertheless underperforming VIPaint-2. We show imputations for multiple test examples in Fig. 3, and see that VIPaint consistently produces plausible inpaintings, while other methods fail to meaningfully inpaint large masks. Note that previous work (Song et al., 2024) has discussed the poor performance of PSLD.

VIPaint yields multiple plausible imputations for large masks. We compare VIPaint with the best performing baseline, CoPaint, across multiple sample inpaintings in Fig. 4. We observe that VIPaint produces multiple visually-plausible imputations while never violating consistency with the observed pixels. These diverse inpaintings also have better quantitative performance than baselines, see Fig. 5. We show diverse imputations using different class conditioning using VIPaint in the Appendix.

VIPaint is effective for text-conditioned foundation models. Results using Stable Diffusion (SD) show that our approach remains robust and effective

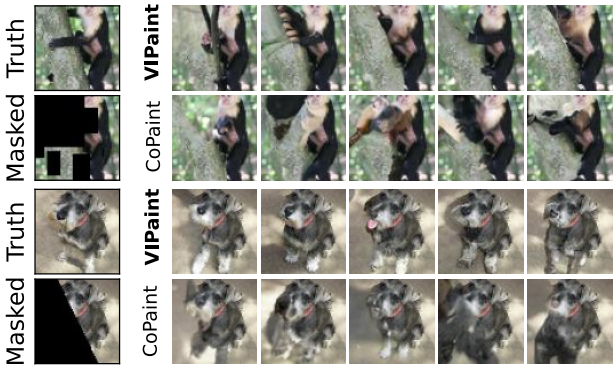

Figure 4: Sample completions comparing VIPaint with the best performing baseline, CoPaint, for two test images. We show 5 inpainted samples for each method. VIPaint yields coherent samples while capturing uncertainty in the missing pixels in images. In contrast, CoPaint has high variance in the quality of results.

even when integrated with state-of-the-art diffusion architectures. VIPaint consistently outperforms DPS on text-conditioned inpainting, producing images that are not only globally coherent but that also preserve fine-grained details. Moreover, VIPaint scales effectively to high-resolution 1024×1024 images. In some cases, the results exceed specialized inpainting models such as SD-Inpainting (Rombach et al., 2022) and Control-

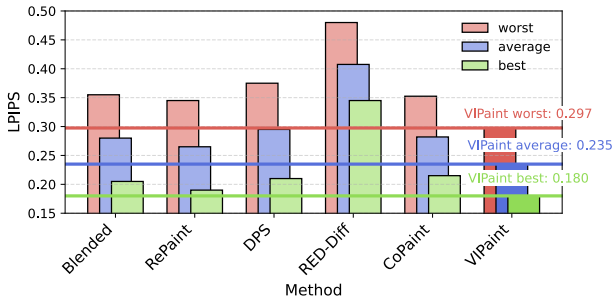

Figure 5: We compute summary statistics (minimum, mean, maximum) of the LPIPS score (lower is better) across *100* sampled completions of each test image. We show the average value of each of these statistics across the full test set. We see that VIPaint improves on all baselines, both in terms of the average quality of results *and* in the consistency in result quality.

Net (Zhang et al., 2023b), which were trained specifically for the inpainting task at greater computational cost. Some results can be found in Fig. 1, and a more comprehensive comparison is in the Appendix.

Computational Efficiency. Table 4 reports the time taken by various methods to produce 10 inpaintings for one test image. REDdiff is fast but inconsistent, and unsuitable for LDMs. Sampling methods are slower and still produce inconsistent results. VIPaint-2 is faster than sampling-based methods for all DMs, *and* achieves better results (see Table 1).

Limitations. VIPaint inherits biases, both good and bad, from whatever DM it is applied to. Like other algorithms for inference with DMs, VIPaint's variational optimization has computational overhead compared to conditional models specialized to inpainting, but this is balanced by its "zero shot" capacity to adapt to new distortion models without expensive retraining.

4.2 Hyperparameters

Keypoints (K) in VIPaint's posterior. Fig. 6 demonstrates the critical importance of using a hierarchical posterior. Removing either keypoint from the VIPaint-2 posterior results in substantially degraded results. Note that this K=1 model is similar to REDdiff, but with a variational distribution defined at $t > 0$. Conversely, the hierarchical VIPaint-2 effectively captures both global and local details.

We find that increasing the number of keypoints in the hierarchy typically improves results for more challenging settings, such as large-mask inpainting on Imagenet256. In this case, VIPaint-4 ($K = 4$) improves average LPIPS from 0.392 to 0.358 for rotated window masks, and from 0.409 to 0.373 for random masks, relative to VIPaint-2. Increasing K has quickly diminishing returns as each update becomes more expensive and optimization becomes more complex, requiring more iterations to converge. For $K > 2$, we find that up-weighting the KL-divergence terms of the loss, similar to the β-VAE (Higgins et al., 2017), can help speed and stabilize convergence while encouraging solutions with higher variance. In general, we recommend $K = 2$ for most settings, as this 2-level posterior can be optimized in as low as 50 iterations and tackles the difficult problem of large mask inpainting well.

Table 4: Runtime Comparison For Inference Methods.

Dataset	**Blended**	**DPS**	**VIPaint**	***Sample***
ImageNet64	(1.13, 1000)	(2.55, 1000)	(1.5, 150)	(1.8, 700)
ImageNet256	(4, 1000)	(10, 500)	(2, 150)	(8, 400)
LSUN	(1.3, 1000)	(5.1, 500)	(2.1, 150)	(4.3, 400)

The (*time in minutes*, *neural function evaluations*) are reported for EDM (top) and LDM (bottom) priors. For VIPaint, optimization ("VIPaint") and sampling are separated, since optimized posterior can be reused. REDdiff matches Blended, while RePaint (2.8 mins) and CoPaint (2.6 mins) are slightly slower than DPS.

Choosing $h(1), h(K)$. We chose the endpoints of VIPaint's posterior based on qualitative analysis on a few validation images, and fixed these values for all experiments (except where noted). We found that spreading keypoints across signal-to-noise ratios $(\alpha_t^2/\sigma_t^2) \in [0.2, 0.5]$ led to good results across models and datasets, concentrating posterior inference on the noise levels which are most crucial to perceptual image quality. The Appendix has further details, and Fig. 6 illustrates robustness to this hyperparameter.

4.3 Deblurring and Superresolution Results

While VIPaint is primarily motivated by the challenge of inpainting large regions with diffusion models, it is a general-purpose inference algorithm applicable to other inverse problems. We consider two linear inverse problems that are widely used as benchmarks, Gaussian deblurring and superresolution. For Gaussian deblurring, we use a kernel with size 61×61 with standard deviation 3.0. For superresolution, we use bicubic downsampling, and a similar experimental setup as Chung et al. (2023).

We compare the performance of VIPaint with ReSample, PSLD, and DPS for ImageNet256 LDM prior. For the pixel-based ImageNet64 EDM model, we compare the VIPaint Gaussian Deblurring performance to DPS. We report the PSNR and LPIPS metrics in Table 5, and give examples in Fig. 7, 8 and the Appendix. We see that for complex image datasets like ImageNet, VIPaint shows strong advantages over DPS and PSLD, and performs similarly to ReSample.

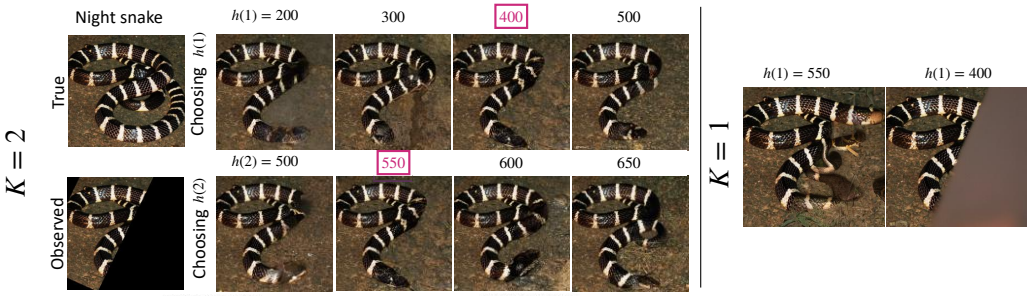

Figure 6: Sensitivity to keypoint selection for the Imagenet256 LDM. *Left:* We fix either $h(1) = 400$ or $h(K) = 550$ and vary the other endpoint around the chosen value to demonstrate robustness. *Right:* A non-hierarchical variant of VIPaint with $K = 1$, for either $h(1) = 400$ or $h(1) = 550$, is inferior to VIPaint-2.

Figure 7: Qualitative examples of superresolution reconstruction with Imagenet256 LDMs. We see that DPS produces extremely blurry images, an artifact that ReSample only partially improves. In contrast, VIPaint-2 leads to samples closer to the true image and produces *very* realistic images.

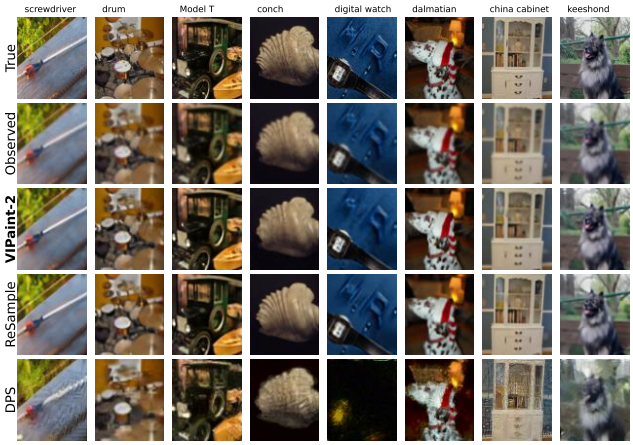

Figure 8: Qualitative examples of removing Gaussian blur with Imagenet256 LDMs. VIPaint again produces sharper results with fewer artifacts than baselines methods.

Table 5: Superresolution and Deblurring Results.

Task	ImageNet256 Super-resolution 4x		ImageNet256 Gaussian Deblur		ImageNet64 Gaussian Deblur	
Metric	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑
VIPaint	**0.37**	**18.90**	0.45	17.91	**0.31**	**13.60**
ReSample	0.40	18.41	**0.44**	**18.03**	–	–
PSLD	0.67	7.77	0.58	0.02	–	–
DPS	0.58	12.99	0.60	12.61	0.32	13.43

Quantitative results (LPIPS, PSNR) for solving linear inverse problems on ImageNet256 using LDM priors, and ImageNet64 using EDM priors. Best results are in bold, and second best results are underlined.

5 CONCLUSION

VIPaint is a principled and general approach to adapt pretrained DMs for image inpainting and other inverse problems. We take widely used (latent) diffusion generative models, allocate variational parameters for the latent codes of each partial observation, and fit the parameters stochastically to optimize the induced variational bound. The simple but flexible structure of our bounds allows efficient VIPaint optimization to outperform previous sampling and variational methods, even for high-resolution text-conditioned LDMs.

Acknowledgements

This research was supported in part by NSF Robust Intelligence Award No. IIS-1816365, ONR Award No. N00014-23-1-2712, and the HPI Research Center in Machine Learning and Data Science at UC Irvine.

References

Agarwal, S., Hope, G., Younis, A., and Sudderth, E. B. (2023). A decoder suffices for query-adaptive variational inference. In Evans, R. J. and Shpitser, I., editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 33–44.

Avrahami, O., Lischinski, D., and Fried, O. (2022). Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218.

Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. (2018). Demystifying MMD GANs. In *International Conference on Learning Representations*.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

Cardoso, G., Janati, Y., Corff, S. L., and Moulines, E. (2024). Monte Carlo guided denoising diffusion models for bayesian linear inverse problems. In *The Twelfth International Conference on Learning Representations*.

Choi, Y., Vergari, A., and Van den Broeck, G. (2020). Probabilistic circuits: A unifying framework for tractable probabilistic models. Technical report.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*.

Chung, H., Sim, B., Ryu, D., and Ye, J. C. (2022a). Improving diffusion models for inverse problems using manifold constraints. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Chung, H., Sim, B., and Ye, J. C. (2022b). Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12413–12422.

Chung, H., Ye, J. C., Milanfar, P., and Delbracio, M. (2024). Prompt-tuning latent diffusion models for inverse problems.

Corneanu, C., Gadde, R., and Martinez, A. M. (2024). Latentpaint: Image inpainting in latent space with diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4334–4343.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255.

Dhariwal, P. and Nichol, A. Q. (2021). Diffusion models beat GANs on image synthesis. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.

Dou, Z. and Song, Y. (2024). Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., and Rombach, R. (2024). Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*.

Feng, B. T., Smith, J., Rubinstein, M., Chang, H., Bouman, K. L., and Freeman, W. T. (2023). Score-based diffusion models as principled priors for inverse imaging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10520–10531.

Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W. T. (2006). Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787–794.

Graikos, A., Malkin, N., Jojic, N., and Samaras, D. (2022). Diffusion models as plug-and-play priors. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851.

Ji, G., Hughes, M. C., and Sudderth, E. B. (2017). From patches to images: A nonparametric generative model. In *International Conference on Machine Learning*, pages 1675–1683.

Kadkhodaie, Z. and Simoncelli, E. (2021). Stochastic solutions for linear inverse problems using the prior

implicit in a denoiser. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 13242–13254.

Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*.

Kawar, B., Elad, M., Ermon, S., and Song, J. (2022). Denoising diffusion restoration models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*.

Kim, J., Park, G. Y., Chung, H., and Ye, J. C. (2025). Regularization by texts for latent diffusion inverse solvers. In *The Thirteenth International Conference on Learning Representations*.

Kingma, D., Salimans, T., Poole, B., and Ho, J. (2021a). Variational diffusion models. In *Advances in Neural Information Processing Systems*.

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. (2021b). On density estimation with diffusion models. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.

Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755.

Liu, A., Niepert, M., and den Broeck, G. V. (2024). Image inpainting via tractable steering of diffusion models. In *The Twelfth International Conference on Learning Representations*.

Liu, X., Gong, C., and Liu, Q. (2022). Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Gool, L. V. (2022). Repaint: Inpainting using denoising diffusion probabilistic models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11451–11461.

Mardani, M., Song, J., Kautz, J., and Vahdat, A. (2024). A variational perspective on solving inverse problems with diffusion models. In *The Twelfth International Conference on Learning Representations*.

Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. (2022). SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*.

Moufad, B., Janati, Y., Bedin, L., Durmus, A. O., Moulines, E., Olsson, J., et al. (2025). Variational diffusion posterior sampling with midpoint guidance. In *The Thirteenth International Conference on Learning Representations*.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. (2022). Glide: Towards photorealistic image generation and editing with text-guided diffusion models.

Nichol, A. Q. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171.

Peebles, W. and Xie, S. (2023). Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241.

Rout, L., Raoof, N., Daras, G., Caramanis, C., Dimakis, A., and Shakkottai, S. (2023). Solving linear inverse problems provably via posterior sampling with latent diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. (2022). Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265.

Song, B., Kwon, S. M., Zhang, Z., Hu, X., Qu, Q., and Shen, L. (2024). Solving inverse problems with

latent diffusion models via hard data consistency. In *The Twelfth International Conference on Learning Representations*.

Song, J., Meng, C., and Ermon, S. (2021a). Denoising diffusion implicit models. In *International Conference on Learning Representations*.

Song, J., Vahdat, A., Mardani, M., and Kautz, J. (2023). Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32.

Song, Y., Shen, L., Xing, L., and Ermon, S. (2022). Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021b). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Spagnoletti, A., Prost, J., Almansa, A., Papadakis, N., and Pereyra, M. (2025). Latino-pro: Latent consistency inverse solver with prompt optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19597–19607.

Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., and Lempitsky, V. (2022). Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159.

Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. S. (2023). Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations*.

Vahdat, A., Kreis, K., and Kautz, J. (2021). Score-based generative modeling in latent space. In *Neural Information Processing Systems (NeurIPS)*.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305.

Wang, H., Yu, Y., Luo, T., Fan, H., and Zhang, L. (2024). MaGIC: Multi-modality guided image completion. In *The Twelfth International Conference on Learning Representations*.

Wang, Y., Yu, J., and Zhang, J. (2023). Zero-shot image restoration using denoising diffusion null-space model. In *The Eleventh International Conference on Learning Representations*.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

Yang, Z., Feng, R., Zhang, H., Shen, Y., Zhu, K., Huang, L., Zhang, Y., Liu, Y., Zhao, D., Zhou, J., and Cheng, F. (2024). Lipschitz singularities in diffusion models. In *The Twelfth International Conference on Learning Representations*.

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. (2015). LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Yu, J., Wang, Y., Zhao, C., Ghanem, B., and Zhang, J. (2023). Freedom: Training-free energy-guided conditional diffusion model. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhang, B., Chu, W., Berner, J., Meng, C., Anandkumar, A., and Song, Y. (2025). Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20895–20905.

Zhang, G., Ji, J., Zhang, Y., Yu, M., Jaakkola, T., and Chang, S. (2023a). Towards coherent image inpainting using denoising diffusion implicit models. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 41164–41193.

Zhang, L., Rao, A., and Agrawala, M. (2023b). Adding conditional control to text-to-image diffusion models. In *ICCV*.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.

Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Chang, E. I., and Xu, Y. (2021). Large scale image completion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**/No/Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**/No/Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/**No**/Not Applicable]
   We will provide publicly available source code to replicate our results after publication.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/**Not Applicable**]
   (b) Complete proofs of all theoretical results. [Yes/No/**Not Applicable**]
   (c) Clear explanations of any assumptions. [Yes/No/**Not Applicable**]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/**No**/Not Applicable]
   Key hyperparameters are documented in the Appendix, and our code release will provide further experimental details.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**/No/Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**/No/Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**/No/Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**/No/Not Applicable]
   (b) The license information of the assets, if applicable. [Yes/No/**Not applicable**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/**Not applicable**]
   (d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]

A Additional Results

A.1 Conditional Inpainting: Text Conditioned

Using Stable Diffusion 3.5 (Esser et al., 2024), our method remains effective when integrated with state-of-the-art diffusion architectures and scales to high-resolution 1024×1024 images. For text-conditioned inpainting, VIPaint consistently outperforms DPS, yielding images that maintain global coherence and preserve fine details. In several cases, VIPaint also surpasses task-specific inpainting models, including SD-Inpainting (Esser et al., 2024) and ControlNet (Zhang et al., 2023b). Minor artifacts remain in baseline methods, such as boundary blending inconsistencies and repetitive filling of masked regions from the prompt without considering the surrounding context, leading to globally inconsistent results. See Figs. 9, 10 for qualitative examples.

Synthetic images are generated using Stable Diffusion 3.5 at a resolution of 1024×1024. The text prompts used for image generation were produced by ChatGPT using the instruction: "Generate 100 realistic image-generation prompts. Each prompt should describe a single, coherent scene in natural language with photographic details (camera type, lighting, time of day, composition, mood, and subject). Focus on professional photography terms (e.g., long exposure, shallow depth of field, aerial view, macro, golden hour)."

A.2 Linear Inverse Problems

To evaluate VIPaint performance at linear inverse problems other than inpainting, we also consider Gaussian deblurring and superresolution. We compare the performance of VIPaint with ReSample, PSLD & DPS for ImageNet256 dataset using the LDM prior. For the pixel-based model, we include results for Gaussian Deblurring comparing VIPaint with DPS. Some qualitative plots are in Fig. 7, 8 and 11.

A.3 Conditional Inpainting

For the case of large-mask image inpainting, we perform some qualitative experiments where we change the input class condition for a given masked observation into the diffusion generative model as shown in Fig 12. We see that VIPaint generates images consistent with the different input class condition *while* also enforcing consistency with the observed set of pixels.

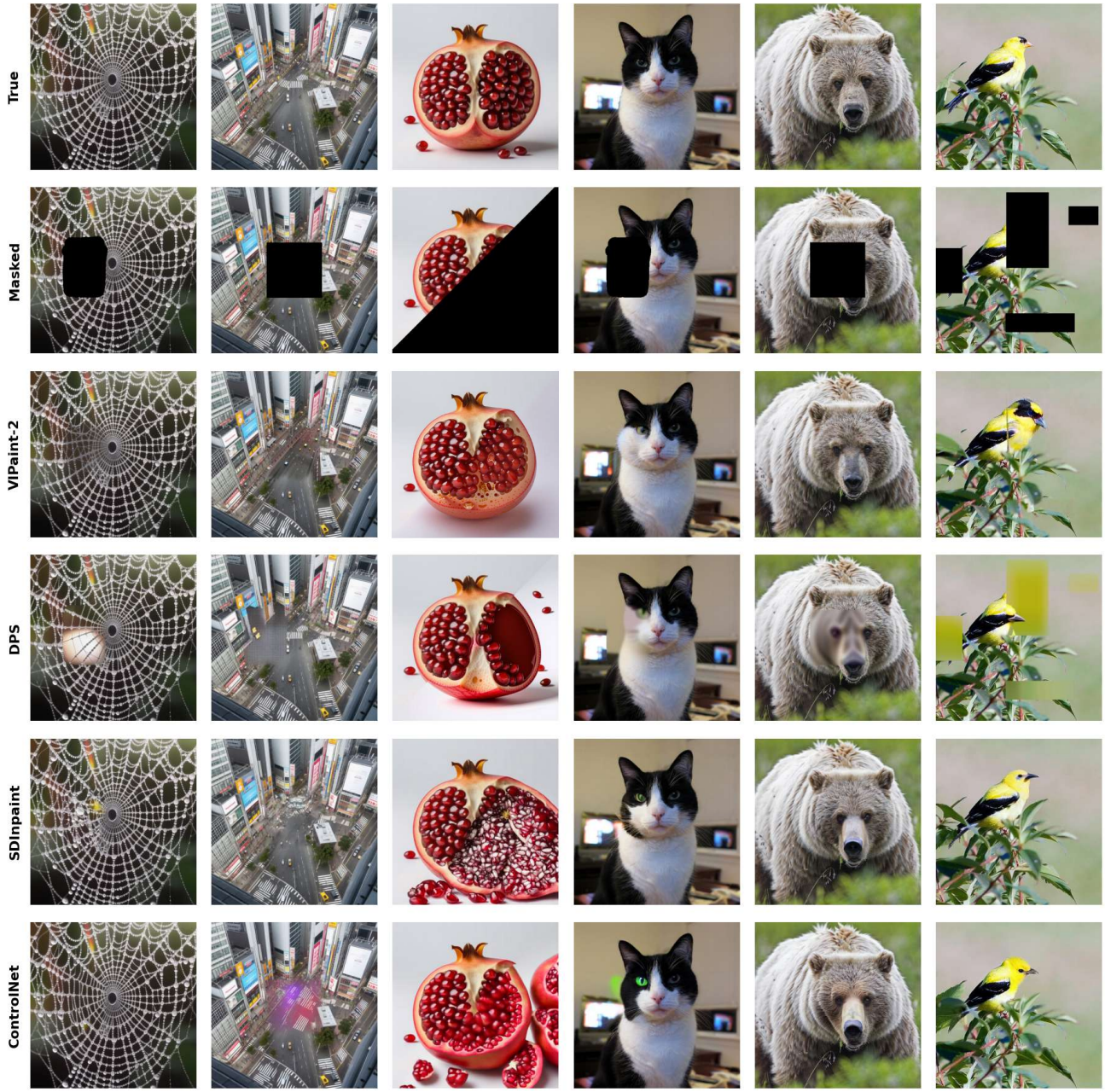

Figure 9: This figure shows image completion results using the LDM prior with the pretrained Stable Diffusion 3.5 turbo model, conditioned on image prompts. Results are reported for synthetic data (left three) and MSCOCO Validation 2014 (right three) under different inpainting masks (Drawn Mask, Center Mask, and Random Masking). While DPS produces blurry reconstructions and the baseline method often fails to remain consistent with the observed image, VIPaint 2.0 successfully captures global semantics and produces realistic inpaintings. Example prompts include: "Close-up of dew-covered spiderweb, rainbow refractions in droplets," "Night cityscape of Tokyo's Shibuya Crossing from rooftop, neon lights on wet pavement," "Close-up of freshly cut pomegranate with glistening seeds," "A black-and-white tuxedo cat with bright green eyes sitting upright indoors, looking directly at the camera in front of a blurred background," "A large brown bear with thick fur walking forward through tall green grass, staring straight ahead," and "A small bright yellow bird with black wings perched on leafy green branches in sunlight, with a soft blurred background."

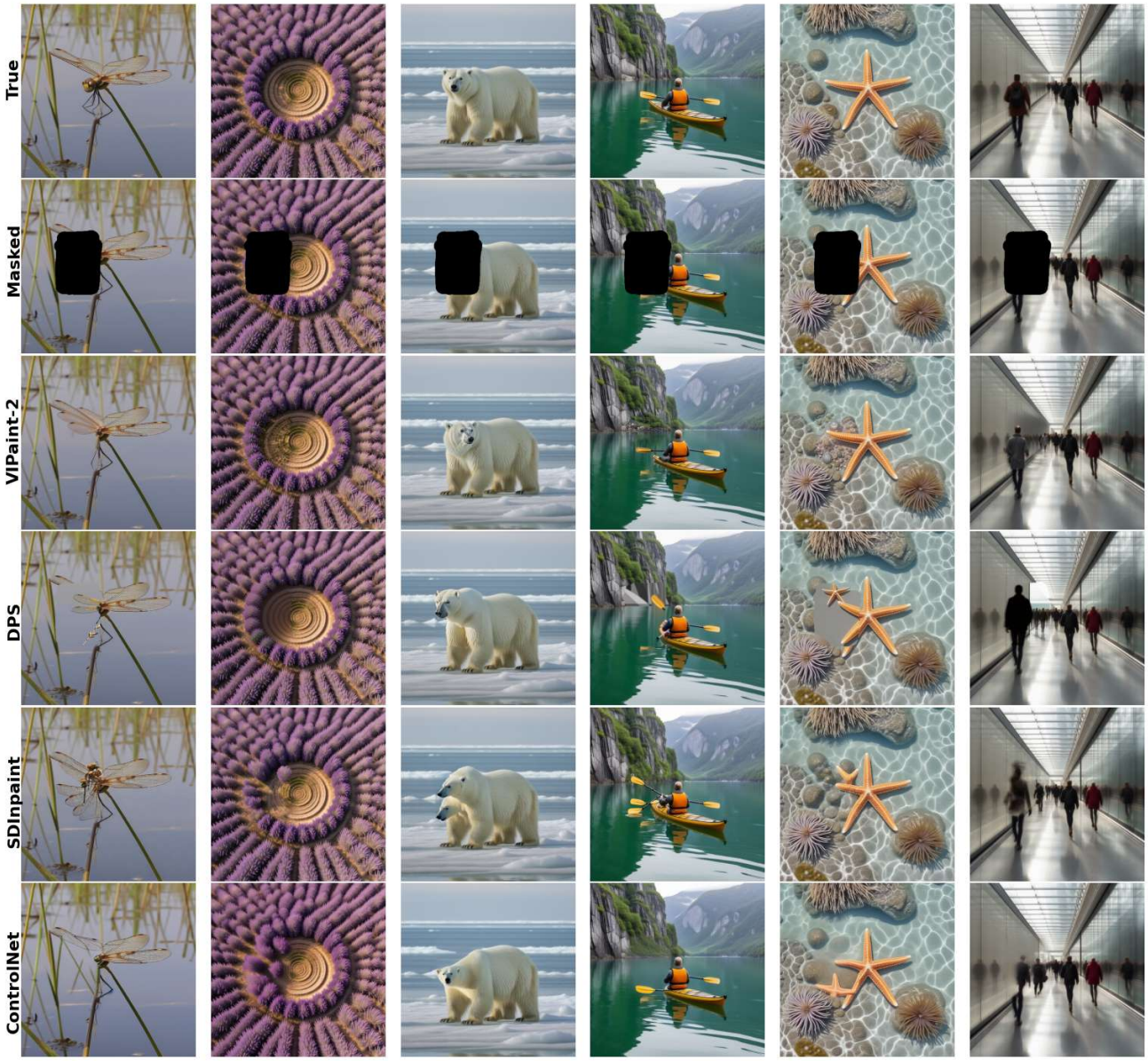

Figure 10: This figure shows image completion results using the LDM prior with the pretrained Stable Diffusion 3.5 turbo model, conditioned on image prompts. Results are reported for synthetic data drawn masks. While DPS produces blurry reconstructions and the baseline method often fails to remain consistent with the observed image, VIPaint 2.0 successfully captures global semantics and produces realistic inpaintings. Prompts used include: "Dragonfly perched on reed above still pond, wings catching light," "Drone shot of a perfectly circular lavender field under the midday sun," "Polar bear standing on sea ice, vast Arctic horizon behind," "Kayaker paddling through a still emerald fjord in Norway, cliffs reflected," "Shallow tide pool with starfish and anemones, clear water surface," and "Long exposure of commuters moving through a glass atrium, ghostly trails."

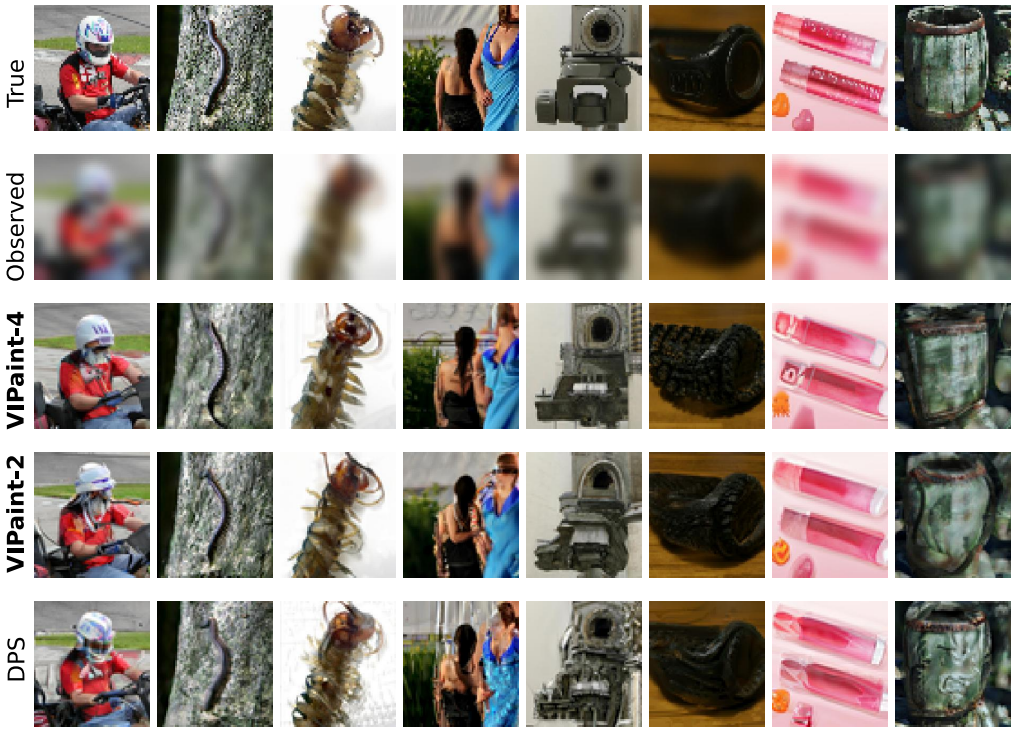

Figure 11: Qualitative results for Gaussian DeBlurring using EDM prior for ImageNet64. We see VIPaint leads to samples closer to the true image and produces *more* realistic images.

B Diffusion Models: Definition & Training Procedure Recap

B.1 Forward time Diffusion Process

The background and expressions on forward diffusion process is taken from Kingma et al. (2021b) and included here for completeness. Re-iterating $q(z_t \mid \bar{x} = \mathbf{enc}(x)) = \mathcal{N}(z_t \mid \alpha_t \bar{x}, \sigma_t^2 I)$, we have the forward diffusion as:

$$q(z_t \mid x) = \mathcal{N}(\alpha_t x, \sigma_t^2 I). \tag{11}$$

Forward Conditional $q(z_t|z_s)$: The distribution $q(z_t|z_s)$ for any $t > s$ are also Gaussian, and from Kingma et al. (2021b), we can re-write as

$$\mathcal{N}(\alpha_{t|s} z_s, \sigma_{t|s}^2 I) \tag{12}$$

$$\text{where,} \quad \alpha_{t|s} = \alpha_t / \alpha_s, \tag{13}$$

$$\text{and,} \quad \sigma_{t|s}^2 = \sigma_t^2 - \alpha_{t|s}^2 \sigma_s^2 \tag{14}$$

Reverse Conditional, $q(z_s|z_t, x)$: The posterior $q(z_s|z_t, x)$ from Kingma et al. (2021b) can be written as:

$$q(z_s|z_t, x) = \mathcal{N}(\mu_Q(z_t, x; s, t), \sigma_Q^2(s, t) I) \tag{15}$$

$$\text{where,} \quad \sigma_Q^2(s, t) = \sigma_{t|s}^2 \sigma_s^2 / \sigma_t^2 \tag{16}$$

$$\text{and,} \quad \mu_Q(z_t, x; s, t) = \frac{\alpha_{t|s} \sigma_s^2}{\sigma_t^2} z_t + \frac{\alpha_s \sigma_{t|s}^2}{\sigma_t^2} x \tag{17}$$

B.2 Reverse Diffusion: Defining $p_\theta(z_s|z_t)$

Here, we describe in detail the conditional reverse model distributions $p_\theta(z_s|z_t)$ for the two cases of variance-exploding and variance preserving diffusion process. Given these formulations, it is straightforward to compute the KL distance between our posterior $q_\lambda(z_s|z_t, y)$ and the prior $p_\theta(z_s|z_t)$ in our loss objective (Eq. 8) since both

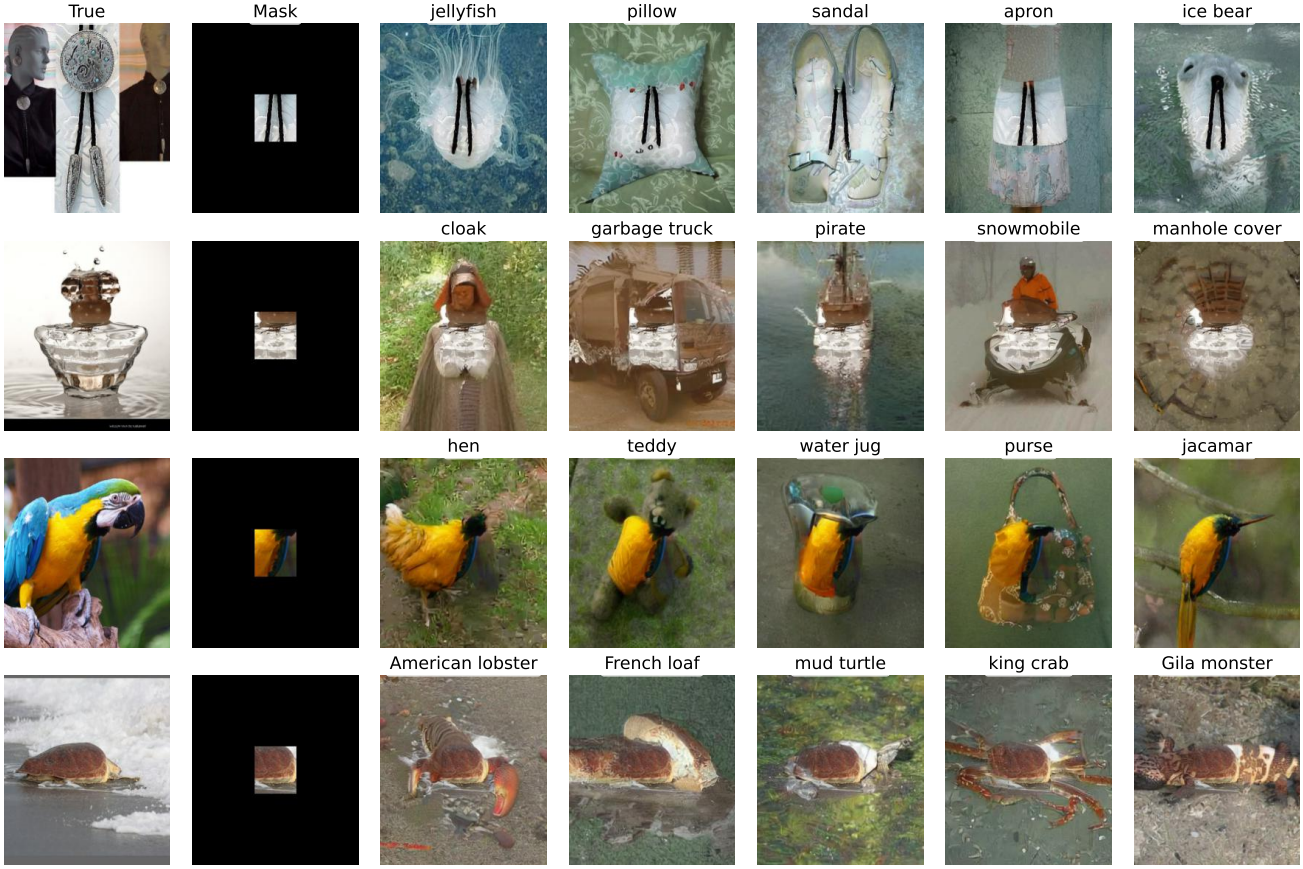

Figure 12: Qualitative results for VIPaint diversity for ImageNet256 with LDM prior using different class conditioning. We see that VIPaint follows the input label and ensures consistency with the observed set of pixels.

are conditionally Gaussian distributions and computing the KL between two Gaussians can be done in closed form.

Variance Exploding Diffusion Process. In this case, $\alpha_t = 1$ and σ_t is usually in the range $[0.002, 50]$ Song et al. (2021b). We follow the ancestral sampling rule from the same work to define our prior conditional Gaussian distributions $p_\theta(z_s|z_t)$:

$$p_\theta(z_s|z_t) = \mathcal{N}(\mu_\theta(z_t; s, t), \sigma_Q^2(s,t)I) \tag{18}$$

$$\text{where,} \quad \sigma_Q^2(s,t) = (\sigma_t^2 - \sigma_s^2)\frac{\sigma_s^2}{\sigma_t^2} \tag{19}$$

$$\text{and,} \quad \mu_\theta(z_t; s, t) = \frac{\sigma_s^2}{\sigma_t^2} z_t + \frac{\sigma_t^2 - \sigma_s^2}{\sigma_t^2}\hat{x}_\theta(z_t, t) \tag{20}$$

where $\hat{x}_\theta(z_t, t) = z_t - \sqrt{(\sigma_t^2 - \sigma_s^2)} * \epsilon_\theta(z_t, t)$

Variance Preserving Diffusion Process. In this case, $\alpha_t = \sqrt{1 - \sigma_t^2}$ and σ_t^2 is usually in the range $[0.001, 1]$ Ho et al. (2020). We follow the DDIM sampling rule Song et al. (2021a) to define our prior conditional Gaussian distributions $p_\theta(z_s|z_t)$. This sampling rule is widely used to generate unconditional samples in small number of steps, and naturally becomes a key design choice of our prior. Here,

$$p_\theta(z_s|z_t) = \mathcal{N}(\mu_\theta(z_t; s, t), \sigma_Q^2(s,t)I) \tag{21}$$

$$\text{where,} \quad \sigma_Q^2(s,t) = \eta(\frac{1 - \alpha_{t-1}}{1 - \alpha_t})(1 - \frac{\alpha_t}{\alpha_{t-1}}) \tag{22}$$

$$\text{and,} \quad \mu_\theta(z_t; s, t) = \sqrt{\alpha_{t-1}}\hat{x}_\theta(z_t, t) + \sqrt{1 - \alpha_t - \sigma_t^2}\epsilon_\theta(z_t, t) \tag{23}$$

where $\hat{x}_\theta(z_t, t) = \frac{z_t - \sqrt{1-\alpha_t}\epsilon_\theta(z_t,t)}{\sqrt{\alpha_t}}$. This schedule is adopted by the Latent Diffusion models.

Rectified Flow. In this case, $\alpha_t = 1 - t$ and $\sigma_t = t$, corresponding to a linear interpolation between data and noise (Liu et al., 2022). The forward process is given by $z_t = (1 - t)x + t\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. The model is trained to predict the velocity field $v_\theta(z_t, t) \approx \epsilon - x$, from which we can recover the clean data estimate as $\hat{x}_\theta(z_t, t) = z_t - t \cdot v_\theta(z_t, t)$. We follow the Euler sampling rule to define our prior conditional distributions $p_\theta(z_s|z_t)$:

$$p_\theta(z_s|z_t) = \mathcal{N}(\mu_\theta(z_t; s, t),\ \sigma_Q^2(s,t)I) \tag{24}$$

$$\text{where,} \quad \sigma_Q^2(s,t) = 0 \tag{25}$$

$$\text{and,} \quad \mu_\theta(z_t; s, t) = z_t + (s - t) \cdot v_\theta(z_t, t) \tag{26}$$

which corresponds to a deterministic ODE step.

B.3 Derivation of Objective for training Diffusion Models: $\mathcal{L}_{(0,T)}(z_0)$

We begin with the standard formulation of the negative ELBO:

$$\begin{aligned} -\log p_\theta(x) \leq \mathcal{L}(\theta; x) &= \mathbb{E}_{z_{0:T}}\left[\log \frac{q(z_{0:T} \mid x)}{p(x \mid z_0)p_\theta(z_{0:T})}\right] \\ &= \mathbb{E}_{z_{0:T}}\left[\sum_{t=1}^{T} \log \frac{q(z_{t-1} \mid z_t, x)}{p_\theta(z_{t-1} \mid z_t)} + \log \frac{q(z_T \mid x)}{p(z_T)} - \log p(x \mid z_0)\right] \end{aligned}$$

As $p(z_T) \approx q(x_T \mid x) \equiv \mathcal{N}(0, I)$, and $p(x \mid z_0)$ are all fixed, we may consider them a constant for the purposes of optimizing θ

$$\begin{aligned} &= \sum_{t=1}^{T} \mathbb{E}_{z_t, z_{t-1}}\left[\log \frac{q(z_{t-1} \mid z_t, x)}{p_\theta(z_{t-1} \mid z_t)}\right] + C \\ &= T\ \mathbb{E}_{t, z_t}\left[D_{KL}\left[q(z_{t-1} \mid z_t, x) \mid\mid p_\theta(z_{t-1} \mid z_t)\right]\right] + C \end{aligned}$$

Recall that

$$p_\theta(z_{t-1} \mid z_t) = q(z_{t-1} \mid z_t, \bar{x} = \hat{x}_\theta(z_t, t)), \text{where} \quad \hat{x}_\theta(z_t, t) = \frac{z_t - \sigma_t \hat{\epsilon}_\theta(z_t, t)}{\alpha_t}. \tag{27}$$

By definition, these two distributions have equal variance ($\sigma_Q^2(t-1,t)$), differing only by the means, $\mu_Q(t-1,t)$ and $\mu_\theta(t-1,t)$, as define above. Therefore we can re-write the KL-divergence as:

$$\begin{aligned}
D_{KL} &= \frac{1}{2\sigma_Q^2(t-1,t)} ||\mu_Q(z_t; t-1, t) - \mu_\theta(t-1,t)||_2^2 \\
&= \frac{1}{2} \frac{\sigma_t^2}{\sigma_{t|t-1}^2 \sigma_{t-1}^2} || \left(\frac{\alpha_{t|t-1}\sigma_{t-1}^2}{\sigma_t^2} z_t + \frac{\alpha_{t-1}\sigma_{t|t-1}^2}{\sigma_t^2} x \right) - \left(\frac{\alpha_{t|t-1}\sigma_{t-1}^2}{\sigma_t^2} z_t + \frac{\alpha_{t-1}\sigma_{t|t-1}^2}{\sigma_t^2} \hat{x}_\theta(z_t, t) \right) ||_2^2 \\
&= \frac{1}{2} \frac{\sigma_t^2}{\sigma_{t|t-1}^2 \sigma_{t-1}^2} || \frac{\alpha_{t-1}\sigma_{t|t-1}^2}{\sigma_t^2} \left(x - \hat{x}_\theta(z_t, t) \right) ||_2^2 \\
&= \frac{1}{2} \frac{\sigma_{t|t-1}^2}{\sigma_t^2 \sigma_{t-1}^2} ||x - \hat{x}_\theta(z_t, t)||_2^2 \\
&= \frac{1}{2} \frac{\sigma_{t|t-1}^2}{\sigma_t^2 \sigma_{t-1}^2} || \frac{1}{\alpha_t}(z_t - \sigma_t \epsilon) - \frac{1}{\alpha_t}(z_t - \sigma_t \epsilon_\theta(z_t; t)) ||_2^2 \\
&= \frac{1}{2} \frac{\sigma_{t|t-1}^2}{\sigma_t^2 \sigma_{t-1}^2} || \frac{\sigma_t}{\alpha_t}(\epsilon - \epsilon_\theta(z_t; t)) ||_2^2 \\
&= \frac{1}{2} \left(\frac{\sigma_t^2 \alpha_{t-1}^2}{\alpha_t^2 \sigma_{t-1}^2} - 1 \right) ||\epsilon - \epsilon_\theta(z_t; t)||_2^2
\end{aligned}$$

Substituting back into the negative ELBO formulation, we get:

$$-\log p_\theta(x) \leq \mathcal{L}(\theta; x) = \frac{T}{2} \mathbb{E}_{t,\epsilon} \left[\left(\frac{\sigma_t^2 \alpha_{t-1}^2}{\alpha_t^2 \sigma_{t-1}^2} - 1 \right) ||\epsilon - \hat{\epsilon}_\theta(\alpha_t \bar{x} + \sigma_t \epsilon, t)||_2^2 \right] + C. \tag{28}$$

B.4 Existing Posterior Sampling Methods for Inverse Problems

Blending Methods methods (Song et al., 2022; Wang et al., 2023) define a procedural, heuristic approximation to the posterior and is tailored for image inpainting. They first generate unconditional samples z_{t-1} from the prior using the learned noise prediction network, and then incorporate y by replacing the corresponding dimensions with the observed measurements. RePaint (Lugmayr et al., 2022) attempts to reduce visual inconsistencies caused by blending via a resampling strategy. A "time travel" operation is introduced, where images from the current time step z_{t-1} are first blended with the noisy version of the observed image y_{t-1}, and then used to generate images in the $(t-1)+r, (r \geq 1)$ time step by applying a one-step forward process and following the Blended denoising process.

Gradient-Based Methods. Motivated by the goal of addressing more general inverse problems, Diffusion Posterior Sampling (*DPS*) (Chung et al., 2023) uses Bayes' Rule to sample from $p_\theta(z_{t-1}|z_t, y) \propto p_\theta(z_{t-1}|z_t)p_\theta(y|z_{t-1})$. Instead of directly blending or replacing images with noisy versions of the observation, DPS uses the gradient of the likelihood $\log p_\theta(y|z_t)$ to guide the generative process at every denoising step t. Since computing $\nabla_{z_t} \log p(y|z_{t-1})$ is intractable due to the integral over all possible configurations of $z_{t'}$ for $t' < t-1$, DPS approximates $p(y|z_{t-1})$ using a one-step denoised prediction $\hat{x}$ using Eq. (27). The likelihood $p(y|x) = \mathcal{N}(f(x), \sigma_v^2)$ can then be evaluated using these approximate predictions. To obtain the gradient of the likelihood term, DPS require backpropagating gradients through the denoising network used to predict $\hat{x}$.

Specializing to image inpainting, *CoPaint* (Zhang et al., 2023a) augments the likelihood with another regularization term to generate samples z_{t-1} that prevent taking large update steps away from the previous sample z_t, in an attempt to produce more coherent images. Further, it proposes CoPaint-TT, which additionally uses the time-travel trick to reduce discontinuities in sampled images.

Originally designed for pixel-space diffusion models, it is difficult to adopt these works directly to latent diffusion models. Posterior Sampling with Latent Diffusion (*PSLD*) (Rout et al., 2023) first showed that employing *DPS*

directly on latent space diffusion models produces blurry images. It proposes to add another "gluing" term to the measurement likelihood which penalizes samples z_t that do not lie in the encoder-decoder shared embedding space. However, this may produce artifacts in the presence of measurement noise (see Song et al. (2024)). To address this issue, recent concurrent work on the *ReSample* (Song et al., 2024) method divides the timesteps in the latent space into 3 subspaces, and optimizes samples z_t in the mid-subspace to encourage samples that are more consistent with observations. Other work (Yu et al., 2023) highlights a 3-stage approach where data consistency can be enforced in the latter 2 stages which are closer to $t = 0$.

C VIPaint: VI method using Diffusion Models as priors

C.1 Derivation of VIPaint's Training Objective

As specified in the main paper, we define a variational distribution over the latent space variable z as $q_\lambda(z)$ and re-use the diffusion prior to generate $x \sim p_\theta(x \mid z)$. We derive the variational objective here:

$$\begin{aligned}
&\mathcal{L}(\lambda; y) \\
&= \mathbb{E}_{q_\lambda(z,x)}[\log p_\theta(y, x, z) - \log q_\lambda(z, x \mid y)] \\
&= \mathbb{E}_{q_\lambda(z,x)}[\log p_\theta(z) + \log p_\theta(x \mid z_{h(1)}) + \log p_\theta(y \mid z_{h(1)}) - \log q_\lambda(z) - \log q_\lambda(x \mid z_{h(1)})] \\
&= \mathbb{E}_{q_\lambda(z)}[\log p_\theta(y \mid z_{h(1)}) + \log p_\theta(z) - \log q_\lambda(z)] \\
&= \mathbb{E}_{q_\lambda(z)}[\log p_\theta(y \mid z_{h(1)})] - \mathbb{E}_{q_\lambda(z)}[\log q_\lambda(z) - \log p_\theta(z)] \\
&= \mathbb{E}_{q_\lambda(z)}[\log p_\theta(y \mid z_{h(1)})] - \underbrace{\mathbb{E}_{q_\lambda(z)}[\log q_\lambda(z) - \log p_\theta(z)]}_{\text{second term}}
\end{aligned} \tag{29}$$

The second term can be further decomposed as:

$$\begin{aligned}
&= \mathbb{E}_{q_\lambda(z)}[\sum_{i=1}^{K-1} \log q_\lambda(z_{h(i)} \mid z_{h(i+1)}) + \log q_\lambda(z_{h(K)}) - \sum_{i=1}^{K-1} \log p_\theta(z_{h(i)} \mid z_{h(i+1)}) - \log p_\theta(z_{h(K)})] \\
&= \sum_{i=1}^{K-1} \mathbb{E}_{z_{h(i+1)}} D[q_\lambda(z_{h(i)} \mid z_{h(i+1)}) || p_\theta(z_{h(i)} \mid z_{h(i+1)})] - \underbrace{D(q(z_{h(K)}) || p(z_{z_{h(K)}}))}_{\text{diffusion loss}}
\end{aligned} \tag{30}$$

Finally, $\mathcal{L}(\lambda; y)$

$$= \mathbb{E}_{q_\lambda(z)}[\log p_\theta(y \mid z_{h(1)})] - \sum_{i=1}^{K-1} \mathbb{E}_{z_{h(i+1)}} D[q_\lambda(z_{h(i)} \mid z_{h(i+1)}) || p_\theta(z_{h(i)} \mid z_{h(i+1)})] - \underbrace{D(q(z_{h(K)}) || p(z_{z_{h(K)}}))}_{\text{diffusion loss}} \tag{31}$$

Negating the above objective, we get the loss objective as follows.

$$L(\lambda) = \underbrace{-\mathbb{E}_q[\log p_\theta(y \mid z_{h(1)})]}_{\text{reconstruction loss}} + \beta \underbrace{\mathcal{L}_{(h(K),T)}(z_{h(K)})}_{\text{diffusion loss}} + \beta \underbrace{\sum_{i=1}^{K-1} \mathbb{E}_{z_{h(i+1)}} D\Big[q_\lambda(z_{h(i)} \mid z_{h(i+1)}) \mid\mid p_\theta(z_{h(i)} \mid z_{h(i+1)}))\Big]}_{\text{hierarchical loss}}.$$

Now, let's derive the diffusion loss term in the following subsection.

C.2 Derivation of Diffusion Loss for VIPaint

For any $h(K) < s < t < T$, we have :

$$\mathbb{E}_{z_{h(K)} \sim q_\lambda(z_{h(K)})} \left[\log \frac{q(z_{h(K)+1:T} | z_{h(K)})}{p_\theta(z_{h(K):T})} \right] \tag{32}$$

$$= \mathbb{E}_{z_{h(K)} \sim q_\lambda(z_{h(K)})} \left[-\log p(z_T) + \sum_{t \geq h(K)} \log \frac{q(z_t | z_s)}{p_\theta(z_s | z_t)} \right] \tag{33}$$

$$= \mathbb{E}_{z_{h(K)} \sim q_\lambda(z_{h(K)})} \left[-\log p(z_T) + \sum_{t > h(K)} \log \frac{q(z_t | z_s)}{p_\theta(z_s | z_t)} + \log \frac{q(z_{h(K)+1} | z_{h(K)})}{p_\theta(z_{h(K)} | z_{h(K)+1})} \right] \tag{34}$$

$$= \mathbb{E}_{z_{h(K)} \sim q_\lambda(z_{h(K)})} \left[-\log p(z_T) + \sum_{t > h(K)} \log \frac{q(z_s | z_t, z_{h(K)})}{p_\theta(z_s | z_t)} \cdot \frac{q(z_t | z_{h(K)})}{q(z_s | z_{h(K)})} + \log \frac{q(z_{h(K)+1} | z_{h(K)})}{p_\theta(z_{h(K)} | z_{h(K)+1})} \right] \tag{35}$$

$$= \mathbb{E}_{z_{h(K)} \sim q_\lambda(z_{h(K)})} \left[-\log \frac{p(z_T)}{q(z_T | z_{h(K)})} + \underbrace{\sum_{t > h(K)} \log \frac{q(z_s | z_t, z_{h(K)})}{p_\theta(z_s | z_t)}}_{\text{diffusion loss}} - \log p_\theta(z_{h(K)} | z_{h(K)+1}) \right] \tag{36}$$

The first and third term can be stochastically and differentially estimated using standard techniques. Following Kingma et al. (2021b), we derive an estimator for the diffusion loss $\mathcal{L}_{(h(K),T)}(z_{h(K)})$. In the case of finite timesteps $t > h(K)$, this loss is:

$$\sum_{t>h(K)} \mathbb{E}_{q(z_t|z_{h(K)})} D[q(z_s|z_t, z_{h(K)})||p_\theta(z_s|z_t)] \tag{37}$$

Reparametering $z_t \sim q(z_t|z_{h(K)})$ as $z_t = \alpha_{t|h(K)} z_{h(K)} + \sigma_{t|h(K)}\epsilon$, where $\epsilon \sim \mathcal{N}(0,1)$, and to avoid having to compute all $T - h(K)$ terms when calculating the diffusion loss, we construct an unbiased estimator as:

$$\frac{T - h(K)}{2} \mathbb{E}_{\epsilon, t\sim\mathcal{U}(h(K),T)}[D(q(z_s|z_t, z_{h(K)})||p_\theta(z_s|z_t))] \tag{38}$$

where $\mathcal{U}(h(K), T)$ is a uniform distribution to sample $h(K) < t \leq T$ from a non-uniform descretization of timesteps using Karras et al. (2022).

Now, we elaborate on the expression $q(z_s|z_t, z_{h(K)})$ and $p(z_s|z_t)$ for any $h(K) < s < t < T$.

C.2.1 $q(z_s|z_t, z_{h(K)})$

Our posterior at $h(K)$ is $q(z_{h(K)}) = \mathcal{N}(\mu_{h(K)}, \tau^2_{h(K)})$. For any $h(K) < s < t < T$, we have $q(z_s|z_{h(K)}) = \mathcal{N}(\alpha_{s|h(K)} z_{h(K)}, \tau^2_{s|h(K)})$ and $q(z_t|z_s) = \mathcal{N}(\alpha_{t|s} z_s, \sigma^2_{t|s})$, yielding the posterior :

$$q(z_s|z_t, z_{h(K)}) = \mathcal{N}(\mu_Q(z_t, z_{h(K)}; s, t, h(K)), \sigma^2_Q(s, t, h(K))I) \tag{39}$$

$$\text{where,} \quad \sigma^2_Q(s, t, h(K)) = \sigma^2_{t|s} \frac{\tau^2_{s|h(K)}}{\sigma^2_{t|s} + \alpha^2_{s|h(K)}\tau^2_{s|h(K)}} \tag{40}$$

$$\text{and,} \quad \mu_Q(z_t, z_{h(K)}; s, t, h(K)) = \sigma^2_Q \left(\frac{\alpha_{s|h(K)}}{\tau^2_{s|h(K)}} z_{h(K)} + \frac{\alpha_{t|s}}{\sigma^2_{t|s}} z_t \right) \tag{41}$$

$$= \frac{\alpha_{s|h(K)}\sigma^2_{t|s}}{(\sigma^2_{t|s} + \alpha^2_{s|h(K)}\tau^2_{s|h(K)})} z_{h(K)} + \frac{\alpha_{t|s}\tau^2_{s|h(K)}}{(\sigma^2_{t|s} + \alpha^2_{s|h(K)}\tau^2_{s|h(K)})} z_t \tag{42}$$

C.2.2 $p(z_s|z_t)$

The conditional model distributions can be chosen as:

$$p_\theta(z_s|z_t) = q(z_s|z_t, z_{h(K)} = \hat{z}_{\theta,h(K)}(z_t, t)) = \mathcal{N}(z_s; \mu_\theta(z_t, z_{h(K)}; s, t, h(K)), \sigma^2_Q(s, t, h(K))) \tag{43}$$

$$\text{where,} \quad \mu_\theta(z_t, z_{h(K)}; s, t, h(K)) = \frac{\alpha_{s|h(K)}\sigma^2_{t|s}}{(\sigma^2_{t|s} + \alpha^2_{s|h(K)}\tau^2_{s|h(K)})} \hat{z}_{\theta,h(K)}(z_t, t) + \frac{\alpha_{t|s}\tau^2_{s|h(K)}}{(\sigma^2_{t|s} + \alpha^2_{s|h(K)}\tau^2_{s|h(K)})} z_t \tag{44}$$

$$\text{and,} \quad \sigma^2_Q(s, t, h(K)) = \sigma^2_{t|s} \frac{\sigma^2_{s|h(K)}}{\sigma^2_{t|s} + \alpha^2_{s|h(K)}\sigma^2_{s|h(K)}} \tag{45}$$

where $\hat{z}_{\theta,h(K)}(z_t, t) = \frac{z_t - \sigma_{t|h(K)} * \epsilon_\theta(z_t, t)}{\alpha_{t|h(K)}}$

D Expanded Figure

We provide an intuitive plot that compares the VIPaint with existing methods that uses pre-trained diffusion models for image inpainting in Fig. 13.

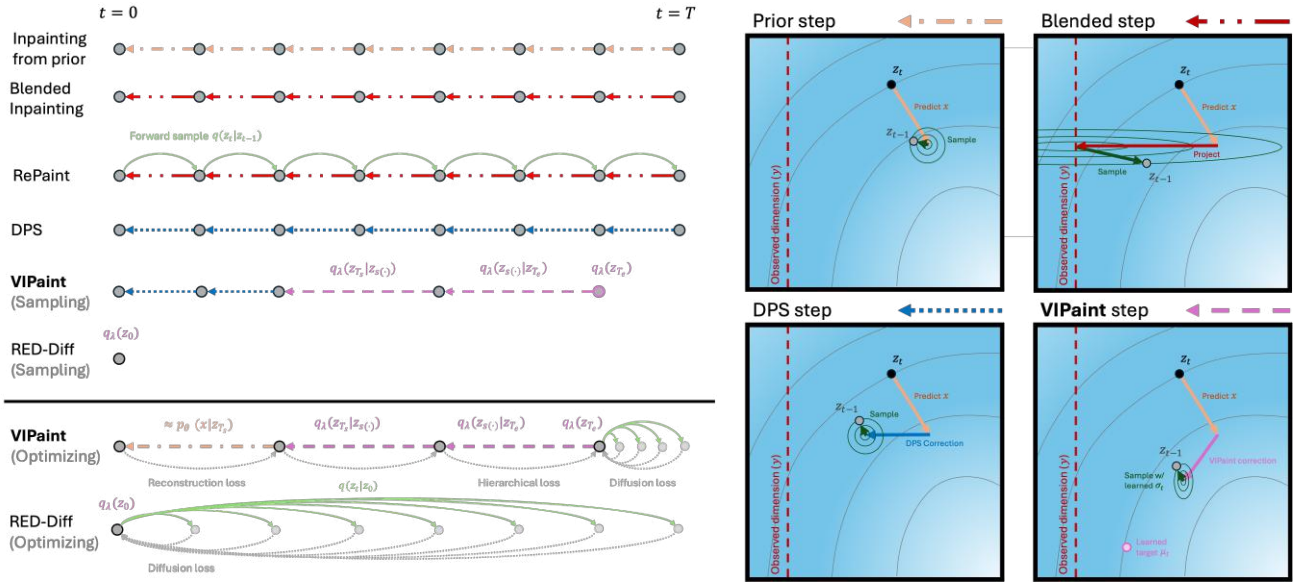

Figure 13: Expanded comparison of methods for diffusion model-based inpainting. **Left:** Timeline illustration of sampling steps with time flowing rightward from $t = 0$ (clean images) to $t = T$ (pure noise). Orange arrows indicate a single step of ancestral sampling under the generative prior $p_\theta(z_{t-1}|z_t)$. Red arrows indicate a single step of the *Blended* approximation of $p_\theta(z_{t-1}|z_t, y)$, while blue arrows indicate a single step of the *DPS* approximation. Green arrows indicate steps forward in time according to the diffusion process $q(z_t|z_{<t})$. Methods such as *RePaint* and *CoPaint* alternate between forward and reverse steps. Purple arrows indicate sampling from a step in the hierarchical **VIPaint** posterior $q_\lambda(z_{s(i-1)}|z_{h(i)})$. Both **VIPaint** and *RED-Diff* (without annealing) involve an initial optimization stage to fit variational parameters per-image. Gray arrows indicate the flow of gradient information during this optimization stage. Gray points are steps only used during optimization. **Right:** Illustration of each reverse-time sampling step in 2 dimensions. The horizontal dimension is assumed to be observed at the value marked by the red line. Each approach begins by computing $p_\theta(z_{t-1}|z_t)$ via a prediction of x using the pre-trained denoising network $\hat{x}_\theta(z_t, t)$. *Blended* replaces observed dimensions with $q(z_{t-1}|y)$. *DPS* updates $p_\theta(z_{t-1}|z_t)$ according to a single-step approximation to the likelihood $p_\theta(y|z_{t-1})$. Finally, **VIPaint**, uses a learned variational distribution $q_\lambda(z_{t-1}|z_t)$, which can be seen as interpolating between the prediction of x and a variational parameter μ_t, coupled with a learned variance.

E Experimental Details

E.1 VIPaint

VIPaint-4 Keeping the two endpoints $h(1), h(K)$ the same, we define the hierarchical posterior over timesteps, $[h(K) = 5, 4, 3.5, 2.5, h(1) = 2]$ for the EDM noise schedule and $[h(K) = 550, 500, 450, h(1) = 400]$ for the LDM prior.

Initialization We follow the forward and reverse diffusion process defined by each VE and VP noise schedules to initialize VIPaint's variational parameters. For LDM prior, we use the lower dimensional encoding of y. We provide a comprehensive summary in Table 6.

Table 6: Initialization of Variational Parameters for VE and VP Schedules

VI Parameters	VP Schedule	VE Schedule
$\mu_{h(K)} = \alpha_{h(K)} y + a_1 \sigma_{h(K)} \epsilon$ (Scale factor to retain information from y.)	$a_1 = 0.8$	$a_1 = 0.01$
$\mu_{h(i)} = \alpha_{h(i)} y + a_2 \sigma_{h(i)} \epsilon$ (Noise adding process is still quite high for VE schedules.)	$a_2 = 1$	$a_2 = 0.01$
$\tau_{h(K)} = \sigma_{h(K)}$ (From the forward diffusion process.)	–	–
$\tau_{h(i)\|h(i+1)}$ (From the reverse diffusion process.)	Eq. 22 with scaling factor a_3/η $a_3 = 0.7$	Eq. 19
$\gamma_{h(i)} \forall i \in [1, K]$ (Weights samples from prior to construct plausible and close to real looking samples.)	0.98 (ImageNet256), 0.88 (LSUN)	0.5

Optimization We fit three sets of variational parameters at every i-th critical time in our hierarchy: means, $\mu_{h(i)}$, variances $\tau^2_{h(i)}$ and weights $\gamma_{h(i)}$. Instead of optimizing τ and γ directly, we optimize the real valued $\tilde{\tau} = \log \tau^2$, and $\tilde{\gamma} = \log(\frac{\gamma}{1-\gamma})$. We optimize this set of variational parameters $\lambda = \{\mu, \tilde{\gamma}, \tilde{\tau}\}$ using Adam with an initial learning rate of $\{0.1, 0.1, 0.01\}$ respectively and decreasing the learning rate by a factor of 0.99 every 10 iterations. We find this setting to be robust across all prior diffusion models and datasets in our work.

During pre-training, most diffusion models parameterize the mean prediction at every diffusion time step t and fix variances, however some previous work (Nichol and Dhariwal, 2021; Dhariwal and Nichol, 2021) has found that (with appropriate training "tricks") learning variances improves performance. Some previous works like ReSample tunes this as a hyperparameter. We instead learn this in our work, and we adjust learning rates to avoid local optima in this process. We optimize the parameters in VIPaint with $K = 2$ for 50 iterations (we show a loss curve in Fig 16); VIPaint with $K = 4$ is optimized for 100 steps in the case of LSUN Churches, 150 steps for the ImageNet64 dataset and 250 steps for the ImageNet256 dataset.

In the case of Stable Diffusion 3.5, a text-conditioned model, we optimized VIPaint with $K = 2$ hierarchical levels for 50 iterations. The corresponding hierarchical bounds were set to $h(1) = 250$ and $h(2) = 500$. We used a learning rate of 0.2 and targeted a resolution of 1024×1024. The training objective from Eq. (8) combines pixel-space reconstruction loss (L_1, weight 0.05), latent-space reconstruction loss (L_2, weight 6.0), diffusion loss (weight 0.5), and hierarchical loss (weight 0.5). We empirically found that this combination, together with a DPS guidance scale of 650.0, provides the best trade-off between reconstruction fidelity and contextual consistency.

Sampling Post training, we take 400 iterative refinement steps from $h(1) = 400$ in the LDM variance preserving schedule to sample inpaintings using a scale factor of 2, similar to the DPS algorithm using perceptual loss. On the other hand, for the EDM prior, we take 700 refinement steps to produce inpaintings after $h(1) = 2$, with scale 5 (similar to DPS tuned for EDM in our work). This scale hyperparameter is tuned over the values [0.1, 0.5, 1, 2, 5, 10] on a validation set of 20 images. During the sampling phase, we use the classifier-free guidance rule with scale = 3 for the ImageNet256 latent diffusion prior.

Reconstruction loss We assume $p(y|z_{h(1)})$ as a Laplacian distribution, where the mean is given by y and a scale parameter, which is computed over 100 images per dataset as a standard deviation over all pixel dimensions. For the 256 pixel datasets, this is 0.56, and for ImageNet64 it is 0.05. In addition to this, we add the perceptual

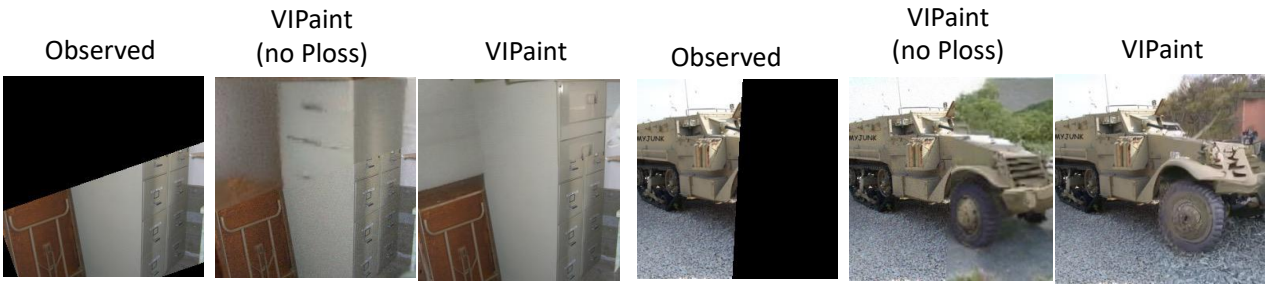

Figure 14: An ablation showing the effect of addition the perceptual loss (PLoss) in the reconstruction term for the task of image inpainting using latent diffusion priors. We see that even though VIPaint can inpaint the image semantically without the Perceptual loss, this loss becomes important to produce sharper reconstructions.

Figure 15: We show the effect of the hyperparameter β with VIPaint with respect to optimization iterations. With $\beta = 10$, VIPaint captures more variations under the diffusion prior instead of "setting" to one kind of completion with $\beta = 1$.

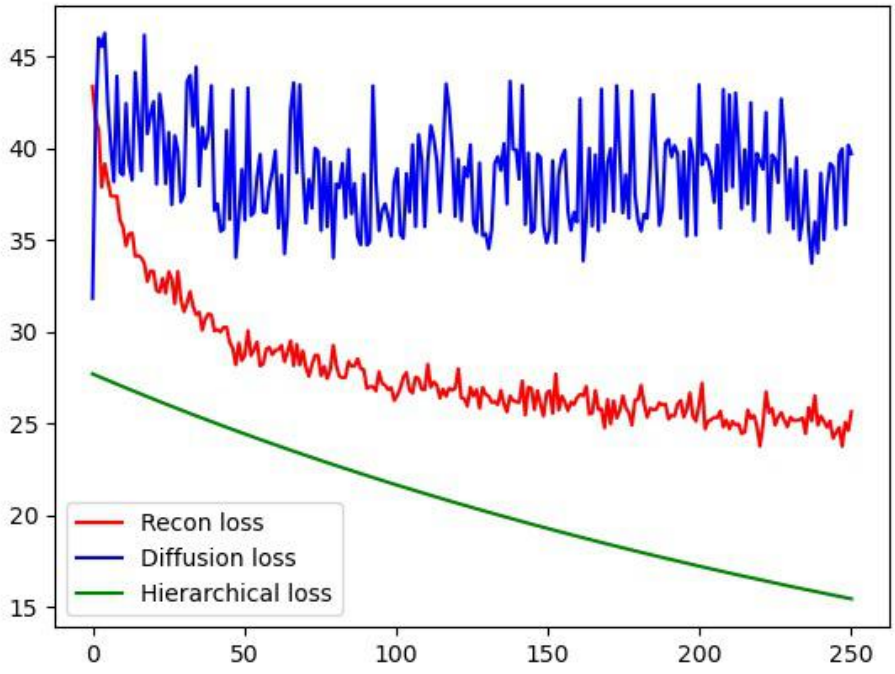

Figure 16: Loss plot over 250 optimization steps for fitting the hierarchical posterior of VIPaint-2.

loss for LDM priors, computing them via feeding the pre-trained Inception network with masked images and masked reconstructions. See Fig. 14 for the benefits of using the perceptual loss with VIPaint.

For VIPaint with $K = 4$, we upweight the KL loss terms in VIPaint's objective with a weight $\beta = 50$ for pixel-based EDM prior and $\beta = 10$ for LDM prior. We show the effect of the different β values in Fig. 15. Generally speaking, higher values of β explores the diffusion latent space more and lower values weighs the likelihood term relatively more and converges faster to a solution.

Descretization of timesteps for prior diffusion loss Lastly, we directly adapt the descretization technique from EDM Karras et al. (2022) to compute the diffusion loss. We use $\rho = 7$ across all models and datasets as used by Karras et al. (2022).

E.2 Baseline Details

Across all the baselines applicable to the latent diffusion models for the ImageNet256 dataset, we use the classifier-free guidance with a scale of 3 (Rombach et al., 2022).

Blended. We run blended for 1000 discretization steps using the EDM and LDM prior.

RePaint. RePaint uses a descritization of 256 steps along with the standard jump length = 10, and number of times to perform this jump operation also set to 10, following standard practice Lugmayr et al. (2022).

DPS. Similar to blended, we take 1000 denoising steps for DPS and set scale = 5 for the edm-based diffusion model, while take 500 steps and keep scale as 0.5 for the Latent Diffusion prior (similar to the original work in Chung et al. (2023)). When using the perceptual loss for the latent diffusion prior, we increase the scale to 2.

PSLD. This is an inference technique only for the Latent Diffusion prior. Similar to DPS, we take 500 steps and keep scaling hyperparameters set to 0.2 as opposed to choosing 0.1 in the original paper Rout et al. (2023). We observe artifacts in the inpainted image if we increase the scale further as also observed by Chung et al. (2024).

CoPaint. We directly adapt the author-provided implementation of CoPaint and CoPaint-TT Zhang et al. (2023a) to use the EDM prior. Apart from the diffusion schedule and network architecture (taken from EDM) all other hyperparameters are preserved from the base CoPaint implementation.

RED-Diff. As with CoPaint, We directly adapt the author-provided implementation of RED-Diff Mardani et al. (2024) and Red-Diff (Var) to use the EDM prior. In this case we increased the prior regularization weight from 0.25 to 50, which we found gave improved performance and more closely matches our VIPaint settings.

ReSample. As with other baselines, we directly adapt the author-provided implementation of ReSample Song et al. (2024) for the LDM prior. Because the original code takes larger optimization steps, resulting in high sampling time, we decrease the number of optimization steps to 50, such that the wall-clock run-time of this method matches the other baselines.

SD-Inpainting. We use the official `StableDiffusionInpaintPipeline` from the Diffusers library with default settings. The model `stabilityai/stable-diffusion-2-inpainting` (Rombach et al., 2022) is used with 50 inference steps to match the optimization steps in our method.

ControlNet. We use the `StableDiffusionControlNetInpaintPipeline` from the Diffusers library with default settings, based on the model proposed by (Zhang et al., 2023b). We run 50 inference steps for consistency with our method.

F Inference Time.

Time Complexity. The time taken by VIPaint-K to optimize a Markov posterior with K keypoints scales primarily with the number of denoising network calls. Each optimization step (assuming $K \ll T$) involves $O(K)$

Table 7: Runtime Comparison For Inference Methods.

Dataset	**Blended**	**DPS**	**VIPaint**	***Sample***
ImageNet64	(1.13, 1000)	(2.55, 1000)	(1.5, 150)	(1.8, 700)
ImageNet256	(4, 1000)	(10, 500)	(2, 150)	(8, 400)
LSUN	(1.3, 1000)	(5.1, 500)	(2.1, 150)	(4.3, 400)

The (*time in minutes*, *neural function evaluations*) are reported for EDM (top) and LDM (bottom) priors. For VIPaint, optimization ("VIPaint") and sampling are separated, since optimized posterior can be reused. RedDiff matches Blended, while RePaint (2.8 mins) and CoPaint (2.6 mins) are slightly slower than DPS.

calls to sample $z_{h(1)} \sim q_\lambda(z_{h(1):h(K)})$, and one to compute the diffusion prior loss, resulting in $O(K)$ function calls per step. Thus, for I optimization steps, the overall complexity is $O(KI)$. For example, our VIPaint-2 is optimized over 50 iterations, it requires only $50 * (2 + 1) = 150$ denoising network calls to infer global image semantics. Once fit, sampling requires an additional $h(1)$ denoising network calls per sample instead of T network calls as in traditional sampling methods.

In tables 7 and 8, we report the time taken for each inference method to produce 10 inpaintings for 1 test image. VIPaint with $K = 2$ is comparable to the baseline methods in terms of wall clock time and the number of functional evaluations (E) of the denoising network. Red-Diff, Blended and RePaint baseline methods do not take gradient of the noise prediction network, whereas all other methods require gradients. In terms of time and number of function calls, we can see that VIPaint-2 takes comparable time and number of function calls as other baselines, but performs far better (Table 9).

Overall, gradient based methods like DPS take longer with an LDM prior because of the use of a decoder per gradient step. PSLD additionally utilizes the encoder and hence, takes longer than DPS.

G Computational resources

All experiments were conducted on a system with 4 Nvidia A6000 GPUs.

Dataset	**VIPaint-2 (opt.)**	**VIPaint-2 (sample)**	**VIPaint-4 (opt.)**	**VIPaint-4 (sample)**
ImageNet64	(1.5, 150)	(1.8, 700)	(10.0, 900)	(1.8, 700)
ImageNet256	(2, 150)	(8, 400)	(10.0, 1250)	(8.0, 400)
LSUN	(2.1, 150)	(4.3, 400)	(5.5, 750)	(4.3, 400)

Table 8: Table comparing (*time in minutes*, *neural function evaluations*) across inference methods using the EDM Prior (top) and LDM Prior (bottom). For VIPaint, we separate the optimization (opt.) and sampling steps, as the optimized posterior approximation can be reused across samples.

Table 9: Quantitative ImageNet64 Inpainting Results.

	Rotated Window			Random Mask		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
VIPaint-2	9.24	**0.56**	**0.30**	**13.33**	**0.62**	**0.23**
CoPaint-TT	8.51	0.51	0.32	12.51	0.58	0.25
CoPaint	8.47	0.50	0.35	12.12	0.56	0.28
RePaint	8.82	**0.56**	0.32	12.05	0.59	0.26
DPS	8.15	0.53	0.32	11.45	0.56	0.29
Blended	7.68	0.52	0.34	11.47	0.57	0.28
RedDiff	8.56	0.45	0.46	11.89	0.51	0.41
RedDiff-V	**9.27**	0.53	0.41	8.35	0.16	0.67

Using the pixel-based EDM prior for all methods, PSNR, SSIM, and LPIPS are averaged over 1000 inpaintings. VIPaint shows the best performance (**bold**), and the second best is underlined.

H Additional Plots

H.1 Analysis of Imagenet Results

Fig. 17 shows details of the comparison between VIPaint and CoPaint with time-travel over 100 randomly selected images from the Imagenet-64 task.

H.2 Small-Mask Image Inpainting for LSUN, ImageNet256

We show some qualitative figures for small masking ratios (upto 20% of the image is corrupted) in Fig. 18 for ImageNet-256 and 19 for LSUN dataset.

H.3 VIPaint captures multi-modal posterior

In addition to producing valid inpaintings, we show multiple samples per test image for all datasets we consider in Fig. 20 -21.

H.4 More qualitative results

We provide more test examples for large mask inpainting in Fig. 22, 23, 24.

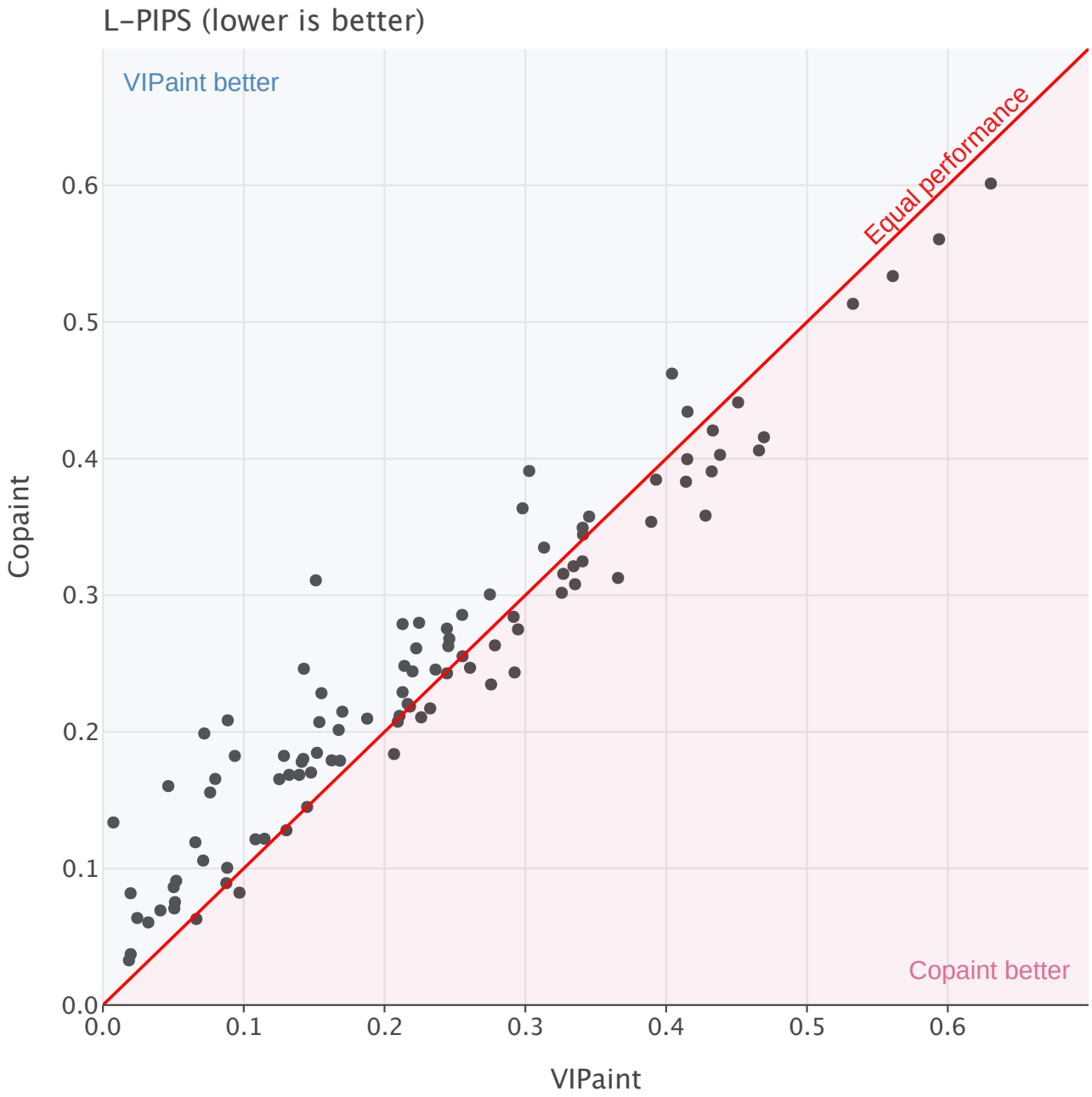

Figure 17: Paired comparison of LPIPS scores for VIPaint-2 and CoPaint with time-travel (CoPaint-TT) on the Imagenet64 "Random Mask" inpainting task (expanding on the experiment shown in table 1. Each point shows the mean LPIPS score across 10 sampled completions of the masked image, with the x and y coordinates showing the VIPaint and CoPaint-TT scores respectively. Additionally, we validated that VIPaint improves on CoPaint-TT using a one-sided paired t-test on the mean LPIPS scores of each method. We found that the improvement was statistically significant with a p-value of **4.133e-05**. As the normality assumption of the t-test may not hold, we also verfied the results using a nonparametric Wilcoxon signed ranked test, which indicated a statistically significant improvement with a p-value of **0.000114**

Figure 18: Qualitative results on the performance across methods for small masking ratios for ImageNet256 dataset using LDM prior. All methods seem to perform reasonably well in this regime.

Figure 19: Qualitative results on the performance across methods for small masking ratios for LSUN dataset using LDM prior. All methods seem to perform reasonably well in this regime.

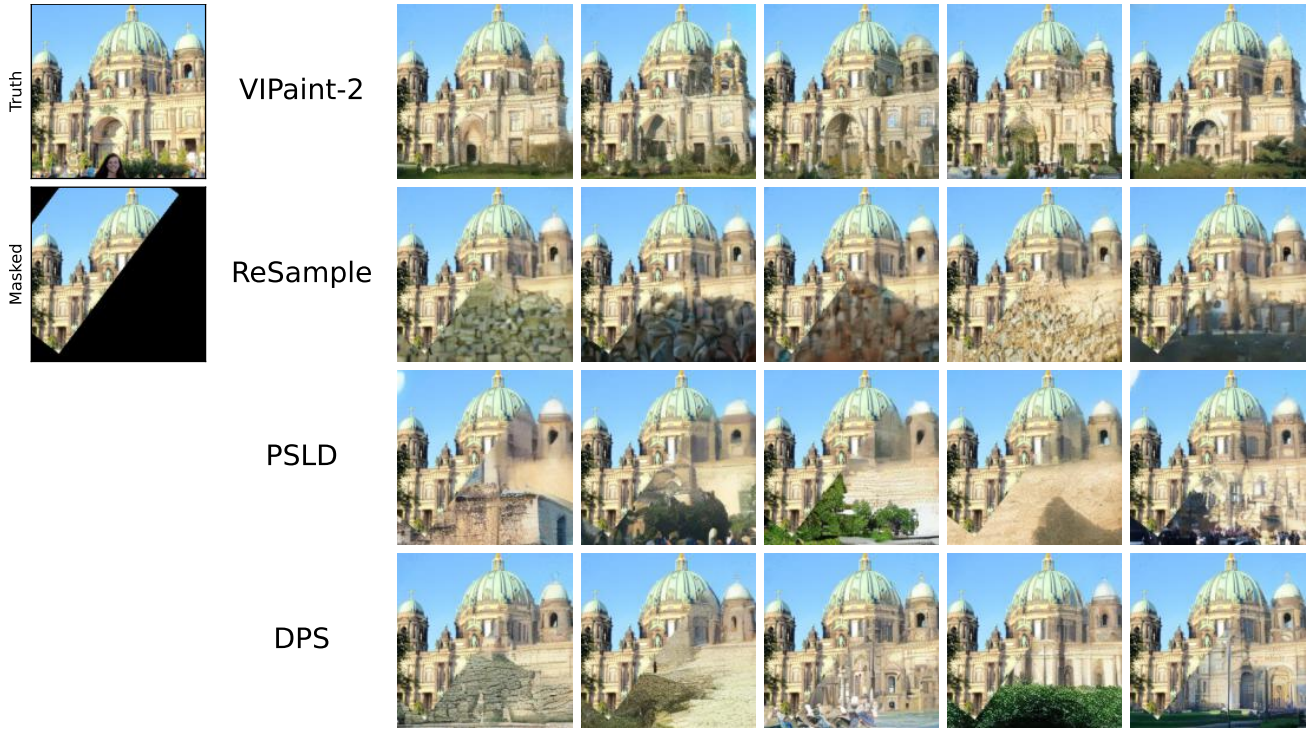

Figure 20: LSUN diversity results. Examples of diverse generation using VIPaint and baseline methods on LSUN using the same input and different initial noise.

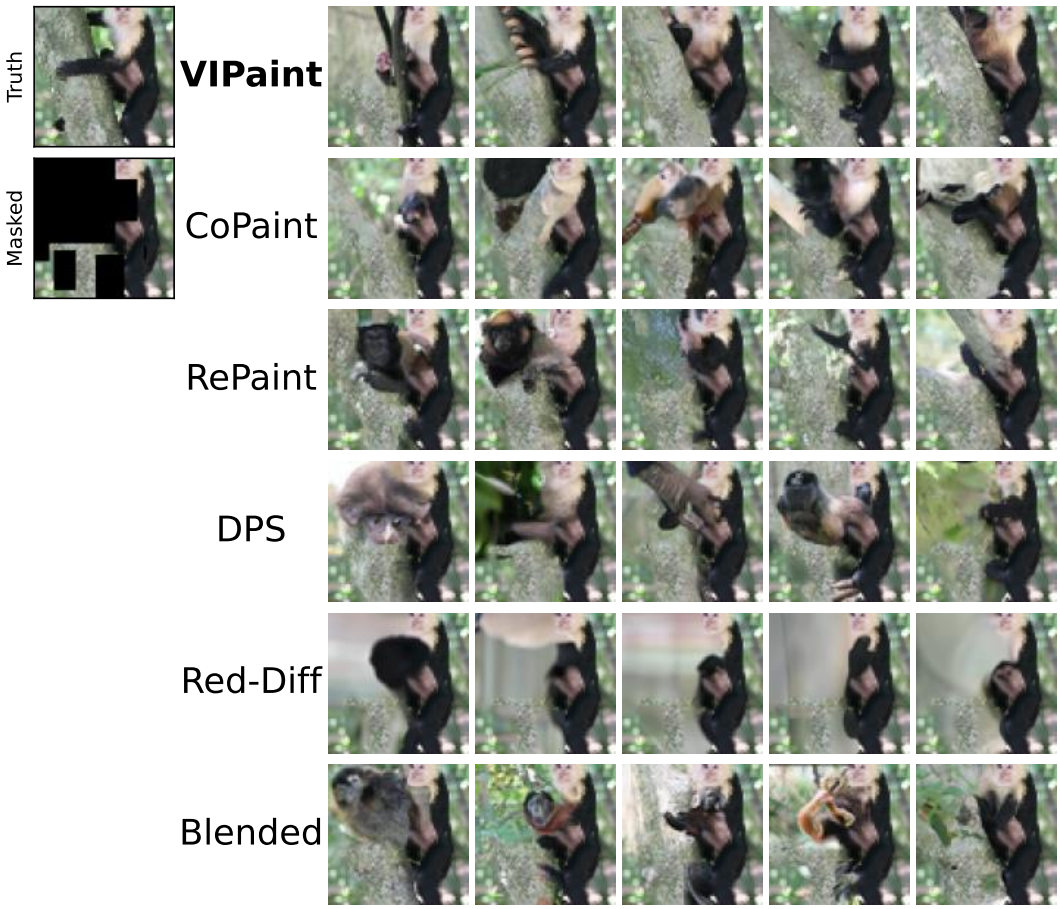

Figure 21: ImageNet64 diversity results with the same class condition but different initial noise.

Figure 22: Image completion results on Imagenet256 using the LDM prior for Rotated Window and Random Masking schemes shown in the second row. We show an inpainting from each method in the following four rows. DPS, PSLD, and ReSample show blurry inpaintings of widely varying quality. In contrast, VIPaint interprets the global semantics in the observed image and produces *very* realistic images. Please find more qualitative plots for LSUN-church in the Appendix Fig. 23.

Figure 23: Qualitative results for LSUN-church dataset using LDM prior for the tasks of image inpainting with large masks. We see that VIPaint-2 can inpaint the images consistently and without any artifacts at the mask borders.

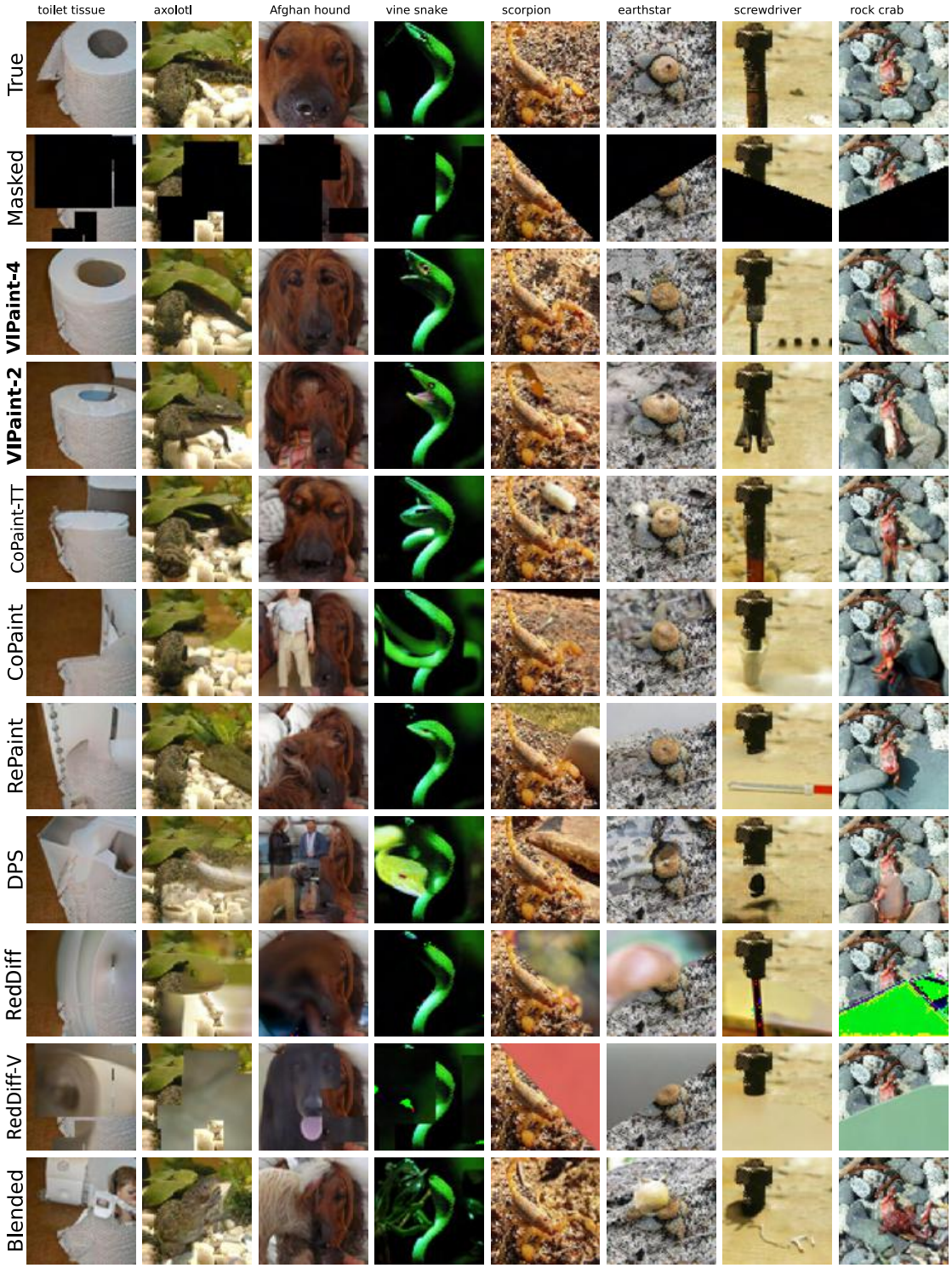

Figure 24: Image completion results on ImageNet64 using a conditional pixel-based EDM prior for image inpainting (Random Masking and Rotated Window schemes) shown in the second row. We show an inpainting from each method in the following rows. Even though the prior diffusion model for ImageNet is conditioned on class labels, inpaintings for baseline methods are inconsistent with the observed image. RePaint and CoPaint is typically more accurate than other baselines, but still produce inconsistent samples unless masks are small. In contrast, VIPaint interprets the global semantics in the observed image while enforcing consistency with the few observed pixels.