# Align & Invert: Solving Inverse Problems with Diffusion and Flow-based Models via Representation Alignment

**Loukas Sfountouris**
Department of Computer Science
University of Warwick, UK
loukas.sfountouris@warwick.ac.uk

**Giannis Daras**
Massachusetts Institute of Technology
Cambridge, MA, USA
gdaras@mit.edu

**Paris Giampouras**
Department of Computer Science
University of Warwick, UK
paris.giampouras@warwick.ac.uk

## Abstract

Enforcing alignment between the internal representations of diffusion or flow-based generative models and those of pretrained self-supervised encoders has recently been shown to provide a powerful inductive bias, improving both convergence and sample quality. In this work, we extend this idea to inverse problems, where pretrained generative models are employed as *priors*. We propose applying *representation alignment* (REPA) between diffusion or flow-based models and a DINOv2 visual encoder, to guide the reconstruction process at inference time. Although ground-truth signals are unavailable in inverse problems, we show that aligning model representations of approximate target features can substantially enhance reconstruction quality and perceptual realism. We provide theoretical results showing (a) that REPA regularization can be viewed as a variational approach for minimizing a divergence measure in the DINOv2 space, and (b) how under certain regularity assumptions REPA updates steer the latent diffusion states toward those of the clean image. We integrate REPA into multiple state-of-the-art inverse problem solvers, and provide extensive experiments on super-resolution, box inpainting, Gaussian deblurring, and motion deblurring confirming that our method consistently improves reconstruction quality, while also reducing the number of discretization steps required to reach the same performance level as the underlying solver. Our code is publicly available at `https://github.com/Sfountouris/Align_And_Invert`, and our project page can be found at `https://sfountouris.github.io/Align_And_Invert/`.

## 1 Introduction

Pretrained diffusion and flow-based models, [36, 15, 24], have recently been at the heart of methods focusing on addressing various types of inverse problems. The crux of these approaches is to perform diffusion sampling, incorporating measurement information during the reverse process, with the goal to reconstruct the clean image at the final step, [29, 6, 39]. Undoubtedly, diffusion and flow-based models have pushed the boundaries in addressing a wealth of inverse problems, [8]. However, they still struggle to produce highly detailed images particularly in cases of severe degradation and in challenging scenes, such as natural images with rich textures and complex backgrounds. These issues are further exacerbated when using latent diffusion models [32], where the inclusion of encoder-

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

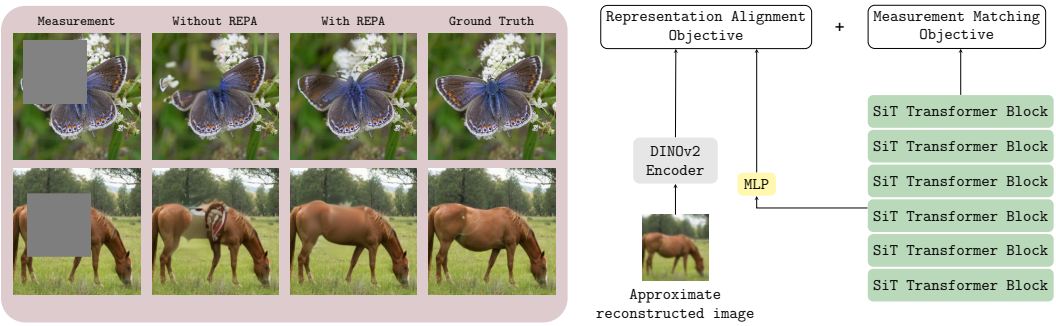



Figure 1: Overview of our proposed framework. Left: Box inpainting results, where adding REPA improves perceptual quality. Right: Alignment between diffusion features and DINOv2 embeddings.

decoder architectures introduces additional challenges during the reverse process due to nonlinearities. To mitigate these limitations, further inductive biases are needed during inference, [33, 31].

Recent studies have shown that diffusion models learn semantic features in their hidden states, [22]. The more expressive these features are the better the diffusion model performs on the generative task, [44]. Building on this insight, the seminal work of [47] introduced a regularizer that aligns the internal representations of the diffusion model with those of a pretrained visual encoder, [27]. This framework, termed *representation alignment* (REPA), was shown to act as a strong semantic constraint leading to higher-fidelity generations and significantly faster convergence.

The success of REPA has attracted considerable interest in this direction, [43, 40, 46, 20, 41]. In particular, [43] observed that while REPA enhances early training dynamics, its benefits diminish over time and can even degrade performance in later iterations. To address this issue, they introduced an early-stopping strategy which mitigated the late-stage degradation and further improved overall convergence speed. Other works have explored applying alignment losses directly to VAE latent spaces [46, 45], finding that diffusion models trained on these regularized spaces converge faster. Despite these advances, most efforts focus on improving training efficiency, whereas relatively little attention has been paid to leveraging representation alignment at inference time, particularly for solving inverse problems. This motivates our central question:

*Can we apply representation alignment to benefit existing algorithms for solving inverse problems using pretrained diffusion and flow-based models?*

**Contributions.** Our main contributions are as follows:

- We introduce a general framework for solving inverse problems with diffusion and flow-based models by enforcing alignment between internal diffusion representations and DINOv2 features (Fig. 1). Despite the absence of ground truth, alignment with proxy reconstructions remains effective thanks to the robustness/invariances of DINOv2 representations.
- We provide theoretical results that characterize the effect of the REPA regularizer: (a) its connection to a divergence measure defined in the DINOv2 feature space, and (b) under suitable regularity assumptions, its ability to steer latent diffusion states toward those associated with the clean image.
- We demonstrate the versatility of our approach by integrating representational alignment into different existing state-of-the-art inverse problem solvers.
- Through extensive experiments on super-resolution, Gaussian deblurring, motion deblurring, and box inpainting, we show consistent improvements over prior methods across tasks and metrics, and further demonstrate that applying REPA to existing solvers allows them to match their original performance using significantly fewer discretization steps.

## 2 Related Work

### 2.1 Diffusion & Flow-based Models

Generative models based on denoising are a class of deep generative models that aim to learn a data distribution $p_0(x)$ by training on progressively corrupted versions of real samples [36, 15, 38,

25]. These models define a) a forward (noising) process, which gradually interpolates between the data distribution and a reference distribution $p_1(x)$ (typically a Gaussian), and b) a reverse (denoising) process, which is trained to invert this corruption and iteratively reconstruct data from noise. Following the formulation of stochastic interpolants [1], the time marginals of this process can be expressed as

$$x_t = \alpha_t x^* + \sigma_t \varepsilon, \qquad x^* \sim p_0(x),\ \varepsilon \sim \mathcal{N}(0, I), \tag{1}$$

where $\alpha_t$ and $\sigma_t$ are decreasing and increasing functions of $t$, respectively, satisfying the boundary conditions $\alpha_0 = \sigma_1 = 1,\ \alpha_1 = \sigma_0 = 0$.. Throughout, we use the notation $\dot{\alpha}_t := \frac{\mathrm{d}\alpha_t}{\mathrm{d}t}$ and $\dot{\sigma}_t := \frac{\mathrm{d}\sigma_t}{\mathrm{d}t}$ to denote time derivatives. It was shown in [24] that there exists a probability flow ODE, $\dot{x}_t = v(x_t, t)$, whose solution follows the same time marginals as the process in eq. (1), where $v(x_t, t)$ denotes the true drift field transporting $p_0(x)$ to $p_1(x)$.

To enable sampling from the data distribution, one has to train a neural network $v_\theta(x_t, t)$ to approximate $v(x_t, t)$. This is done by regressing the network output toward the optimal vector field using *conditional flow matching* [24, 17], minimizing

$$\mathcal{L}_{\text{velocity}}(\theta) := \mathbb{E}_{x^*, \varepsilon, t}\left[\|v_\theta(x_t, t) - \dot{\alpha}_t x^* - \dot{\sigma}_t \varepsilon\|^2\right], \tag{2}$$

Beyond the ODE view, there also exists a reverse *stochastic differential equation* (SDE) with the same time marginals:

$$\mathrm{d}x_t = \left[v(x_t, t) - \frac{1}{2} g^2(t) s(x_t, t)\right] \mathrm{d}t + g(t)\, \mathrm{d}\bar{w}_t. \tag{3}$$

where $s(x_t, t) = \nabla_{x_t} \log p_t(x_t)$ is the score, $g(t)$ is the diffusion coefficient, and $\bar{w}_t$ is a standard Wiener process running backward in time. Notably, the score can be expressed directly in terms of the velocity field:

$$s(x_t, t) = \sigma_t^{-1} \cdot \frac{\alpha_t v(x_t, t) - \dot{\alpha}_t x_t}{\dot{\alpha}_t \sigma_t - \alpha_t \dot{\sigma}_t}. \tag{4}$$

Eq. (4) hints that the score parameterization used in diffusion models and the velocity parameterization used in flow-based models can lead to equivalent formulations of the sampling process. In practice, diffusion models correspond to a stochastic discretization of the probability flow ODE, while flow-based models correspond to its deterministic counterpart [19].

### 2.2 Diffusion Models with Representation Guidance

Diffusion models learn rich, discriminative features in their hidden states, which are crucial for their generative performance [44]. Nevertheless, their learned representations still fall behind on downstream tasks when compared with state-of-the-art self-supervised visual encoders, [3, 35]. This representation gap has been identified as a key bottleneck for improving the generative performance of diffusion models, as richer and more semantically structured representations have been shown to enhance sample fidelity [47]. To address this, a growing line of work introduces representation guidance, which conditions diffusion models on external representations from pretrained self-supervised encoders [23]. Typically these approaches fall into two categories: (i) conditioning the diffusion model on external representations, either learned, [23], or retrieved from a database of samples, [4, 16, 34]; and (ii) explicitly aligning the model's internal representations with those of large pretrained encoders, thereby encouraging more semantically meaningful feature-spaces, [40, 16, 20].

Within the REPA framework, feature-level alignment is applied directly to intermediate representations of the diffusion model. In particular, let the diffusion model consist of $L$ transformer blocks, and $\ell \in \{1, \ldots, L\}$ denote the layer at which features are extracted. The quantity $h_t \in \mathbb{R}^{N \times D_2}$ denotes the matrix of patch tokens obtained after the first $\ell$ transformer blocks when the model processes an input image $x$ at timestep $t$. Here, $N$ is the number of patch tokens and $D_2$ the embedding dimension, with $h_t^{[n]} \in \mathbb{R}^{D_2}$ representing the $n$-th patch token. For convenience, this is written as $h_t = \text{DIFFENC}(x_t, t)$, where $x_t$ denotes the noisy image and DIFFENC returns the patch-level hidden representation after the first $\ell$ blocks of the diffusion model's transformer encoder. REPA enforces patch-wise similarity between the $D_1$-dimensional DINOv2 representation of the target image $x$, i.e., $f_{\text{DINOv2}}^{[n]}(x) \in \mathbb{R}^{D_1}$ for patches $n = 1, \ldots, N$, and the intermediate hidden states $h_t^{[n]}$, by maximizing

the following objective during training:

$$\textsc{Repa}(x, h_t) = \frac{1}{N} \sum_{n=1}^{N} \cos\Big( f_{\text{DINOv2}}^{[n]}(x),\, g_\phi\big(h_t^{[n]}\big) \Big), \tag{5}$$

where $g_\phi : \mathbb{R}^{D_2} \to \mathbb{R}^{D_1}$ is a learnable multi-layer perceptron (MLP) that projects hidden states into the DINOv2 space. By jointly optimizing $\mathcal{L}_{\text{REPA}}$ with the standard diffusion loss, the model learns internal representations that are semantically richer and better aligned with high-level visual features, ultimately yielding higher-quality image generation and faster convergence during training.

### 2.3 Inverse Problems with Diffusion and Flow Models

When solving inverse problems the goal is to recover the clean signal $x_0$ from noisy or degraded measurements $y$. In the flow- and diffusion-based framework, this is accomplished by replacing the unconditional score function in eq. (3) with the conditional score $\nabla_{x_t} \log p(x_t \mid y)$. Applying Bayes' rule, the conditional score can be decomposed as

$$\nabla_{x_t} \log p(x_t \mid y) = \nabla_{x_t} \log p_t(x_t) + \nabla_{x_t} \log p(y \mid x_t),$$

Using this decomposition, the reverse-time SDE becomes

$$dx_t = \big( v(x_t, t) - g^2(t)\, \nabla_{x_t} \log p(x_t \mid y) \big)\, dt + g(t)\, d\bar{w}_t.$$

The unconditional score $\nabla_{x_t} \log p_t(x_t)$ is provided by a pretrained diffusion or flow-based model, while the likelihood term $\nabla_{x_t} \log p(y \mid x_t)$ enforces measurement consistency but is generally intractable, as it requires marginalizing over all possible clean signals $x_0$. One of the most prominent approaches is *Diffusion Posterior Sampling (DPS)*, [6], which uses the following approximation:

$$p(y \mid x_t) \approx p\left(y \mid \hat{x}_0 = \mathbb{E}[x_0 \mid x_t]\right),$$

where $\mathbb{E}[x_0 \mid x_t]$ denotes the conditional expectation of $x_0$ given $x_t$. In many practical scenarios, performing the diffusion process directly in pixel space is computationally prohibitive. Latent diffusion models (LDMs) address this limitation by operating in a compressed latent space. Let $\mathcal{E} : \mathbb{R}^k \to \mathbb{R}^d$ be an encoder mapping an image $x_0$ to its latent representation $z_0 = \mathcal{E}(x_0)$, and let $\mathcal{D}$ denote the corresponding decoder such that $\hat{x}_0 = \mathcal{D}(z_0) \approx x_0$. Building on this formulation, recent work has proposed latent-space counterparts of pixel-based inverse problem solvers. For example, DPS has been formulated in latent space by redefining the measurement likelihood as

$$p(y \mid z_t) \approx p\big(y \,\big|\, \hat{x}_0 = \mathcal{D}\big(\mathbb{E}[z_0 \mid z_t]\big)\big).$$

This straightforward formulation, commonly referred to as Latent DPS, has been observed to introduce artifacts, largely due to the nonlinearity of the decoder. To mitigate this effect, [33] proposed a regularization term encouraging the latent variables to remain close to a fixed point of the encoder–decoder mapping, while [37] enforced data consistency during optimization followed by a projection of the latent variables back onto the noisy manifold.

## 3 Proposed Approach

Given a noisy observation $y \in \mathbb{R}^m$ of an unknown signal $x \in \mathbb{R}^k$, our goal is to sample from $p_\theta(x_0 \mid y)$ using a pretrained diffusion or flow-based models. We aim to apply the REPA framework to inverse problems. However, a key challenge is that representation alignment requires access to the ground-truth signal, which is not available in this setting. To overcome this, we must choose an alternative input to the DINOv2 encoder that can stand in for the missing ground truth. In particular, we need an approximation $\bar{x}_0$ of the unknown $x_0$ that produces a *proxy representation* $c_{\text{proxy}} \in \mathbb{R}^{N \times D_1}$, where $N$ denotes the number of patch tokens and $D_1$ is their feature dimension. This representation should satisfy

$$c_{\text{proxy}} \equiv f_{\text{DINOv2}}(\bar{x}_0) \approx f_{\text{DINOv2}}(x_0). \tag{6}$$

We then introduce a *tilted distribution*, [28], that incorporates representation alignment as reward:

$$\widetilde{p}(x_t \mid y) \propto p(y \mid x_t)\, p_t(x_t) \exp\big(\lambda\, \textsc{Repa}(\bar{x}_0, h_t)\big), \tag{7}$$

where $h_t = \text{DIFFENC}(x_t, t)$ denotes the intermediate diffusion representation and $\lambda > 0$ controls the strength of the alignment term. The likelihood $p(y \mid x_t)$ enforces measurement consistency, while $p_t(x_t)$ corresponds to the unconditional prior induced by the generative model. The REPA term defined in Eq. (5) encourages alignment between the diffusion representation of $x_t$ and the proxy DINOv2 features, thereby biasing the sampling process toward semantically consistent reconstructions.

**On the selection of $c_{\text{proxy}}$.** To address the fact that representation alignment [47] is not directly applicable to inverse problems where the ground-truth image is unavailable at inference time, we construct $c_{\text{proxy}}$ from an approximate reconstruction. We initialize the proxy using the DINOv2 features of the available observation and, as the reverse diffusion process progresses and the reconstruction becomes more informative, we gradually replace this proxy with the features of the model's current denoised estimate, namely $f_{\text{DINOv2}}(\mathbb{E}[x_0 \mid x_t])$. In both cases, we rely on the robustness of pretrained DINOv2 features, which, as shown in our experiments, remain stable under the degradations present in super-resolution and deblurring tasks. An ablation in Appendix A.4 quantifies the reliability of this proxy across tasks. We further compare against alternative encoders, such as CLIP [30], in Appendix A.6, where DINOv2 consistently yields improved performance. We also examine the behavior of the method across a broader range of degradation levels in Appendix A.7, where the proxy-based alignment remains effective even in more challenging settings.

**On the availability of the alignment mapping.** The regularizer in (7) relies on a projection head $g_\phi$ mapping diffusion representations into the DINOv2 feature space. For models trained with representation alignment, this mapping is available from pretraining. More generally, our framework can also be applied to arbitrary pretrained diffusion models by learning a post-hoc mapping between diffusion representations and DINOv2 features while keeping the diffusion model itself frozen. Since only the projection head $g_\phi$ is optimized, this additional training is lightweight and inexpensive, compared to diffusion-model pretraining. We refer the reader to Appendix A.14 for further details. As we show in Appendix A.15, this post-hoc adaptation preserves the benefits of representation alignment for inverse problems and enables REPA regularization to be used with existing generative models not previously pretrained with REPA.

**Extension to latent diffusion models.** The proposed representation-alignment strategy applies naturally to latent diffusion models. Let $z_t$ denote the latent state at timestep $t$. In this setting, the tilted distribution becomes

$$\widetilde{p}(z_t \mid y) \propto p(y \mid z_t) p_t(z_t) \exp\left(\lambda \text{REPA}(\bar{x}_0, h_t)\right),$$

With a slight abuse of notation, we use DIFFENC to denote the diffusion encoder in both pixel-space and latent-space models, and write $h_t = \text{DIFFENC}(z_t, t)$ for latent diffusion.

Our general algorithm for a latent-space velocity-based sampler is summarized in Algorithm 1. Step 6 is intentionally kept abstract; as discussed in Section 2.3, different inverse-problem solvers approximate this term in different ways. The proposed regularizer is compatible with standard diffusion-based reconstruction pipelines.

In our experiments, we instantiate the framework using two state-of-the-art latent diffusion methods, with further implementation details provided in Appendix A.2 and Appendix A.3. We further provide an ablation study in Appendix A.10 analyzing the effect of all hyperparameters of our framework, including the cutoff timestep.

**Algorithm 1** REPA-regularized Inverse Algorithm

**Require:** Flow or diffusion model $u_\theta$, measurement $y$, learning rate $\eta$, regularizer strength $\lambda$, proxy representation $c_{\text{proxy}}$, timestep $t_{\text{cutoff}}$, decoder $\mathcal{D}$. initialize: $z_T \sim \mathcal{N}(0, I)$

1: **for** $t = T, \dots, 1$ **do**
2: $\quad z_{t-1} \leftarrow z_t - \frac{1}{T} u_\theta(z_t, t)$
3: $\quad z_{t-1} \leftarrow z_{t-1} + \eta \nabla_{z_t} \log p(y \mid z_t)$
4: $\quad z_{t-1} \leftarrow z_{t-1} + \lambda \nabla_{z_t} \sum_{n=1}^{N} \cos\left(c_{\text{proxy}}^{[n]}, g_\phi\left(\text{DiffEnc}^{[n]}(z_t, t)\right)\right)$
5: $\quad$ **if** $t < t_{\text{cutoff}}$ **then**
6: $\quad\quad c_{\text{proxy}} \leftarrow f_{\text{DINOv2}}(\mathcal{D}(\mathbb{E}[z_0 \mid z_t]))$
7: $\quad$ **end if**
8: **end for**
9: **return** $z_0$

## 4 Theoretical Results

Recent advances in generative modeling have demonstrated that objectives defined in deep feature spaces, rather than purely pixel-based discrepancies, produce reconstructions that better capture perceptual similarity [11, 49, 13]. Motivated by this observation, our goal in this section is to clarify how the REPA regularizer fits within this broader class of feature-based objectives. In particular, we present two theoretical results that characterize how REPA acts on (i) the DINOv2-space representations of reconstructed images and (ii) the latent diffusion state itself.

First, we show that maximizing the REPA objective encourages alignment between the DINOv2 feature distributions of reconstructed and ground-truth images, yielding an interpretation in terms of maximum mean discrepancy (MMD). Second, we study how REPA affects the latent states of the diffusion model during the reverse process. Under suitable assumptions, we show that a REPA update moves the current diffusion state toward the latent representation of the clean image. This provides a complementary view: REPA influences not only the DINOv2 representation of the reconstruction but also the latent state that guides the sampling trajectory.

Before presenting the main results, we introduce two quantities that capture, respectively, the alignment between feature spaces and the accuracy of the proxy representations used in our analysis.

**Definition 4.1** (Misalignment between DINOv2 and diffusion representations)**.** For an image $x$ and diffusion timestep $t$, we define the representation misalignment as

$$\mathrm{MisREPA}(x,t) := \mathbb{E}_n \, \| f^{[n]}_{\mathrm{DINOv2}}(x) - g_\phi(\mathrm{DiffEnc}^{[n]}(x,t)) \|_2^2,$$

where the expectation is taken uniformly over patch indices $n \in \{1,\dots,N\}$. Here $f^{[n]}_{\mathrm{DINOv2}}(x)$ and $\mathrm{DiffEnc}^{[n]}(x,t)$ denote the $n$-th patch features from the DINOv2 encoder and the diffusion model encoder, respectively. The mapping $g_\phi$ is the projection head introduced in Section 2.2[1].

**Definition 4.2** (Proxy approximation error)**.** For a ground-truth image $x$ and its proxy approximation $\bar{x}$ (defined in Section 3), we define the proxy approximation error as

$$\mathrm{ApproxErr}(x,\bar{x}) := \mathbb{E}_n \left\| f^{[n]}_{\mathrm{DINOv2}}(x) - f^{[n]}_{\mathrm{DINOv2}}(\bar{x}) \right\|_2^2,$$

where the expectation is taken uniformly over patch indices $n \in \{1,\dots,N\}$.

In practice, both quantities defined above are typically small. The feature misalignment $\mathrm{MisREPA}(x,t)$ is mitigated through the explicit alignment of DINOv2 and diffusion encoder representations during training [47], while the approximation error $\mathrm{ApproxErr}(x,\bar{x})$ is reduced by constructing proxy features $\bar{x}$ that serves as reliable substitutes for the unavailable ground-truth features (see Section 3). Robustness of DINOv2 features to common image degradations [27], ensures that the resulting approximation gap remains small across the inverse problems we study.

**Feature-Space Alignment: REPA as an MMD Surrogate.** Building on the definitions introduced above, we now establish a connection between the REPA regularizer and the statistical discrepancy between DINOv2 features of clean and reconstructed images.

**Proposition 4.3.** *Assume $x \sim p_X$ is a ground-truth image and $\hat{x} \sim p_{\hat{X}|Y}$ is its corresponding reconstruction, where $p_{\hat{X}|Y}$ is the distribution induced by any reconstruction method. Assume further that all DINOv2 feature embeddings are $\ell_2$–normalized. Let $\bar{x}$ be a proxy approximation of $x$. Define the empirical mean DINOv2 feature embedding as*

$$\mu_f(x) := \mathbb{E}_n \left[ f^{[n]}_{\mathrm{DINOv2}}(x) \right].$$

*Then the expected REPA alignment satisfies*

$$\begin{aligned}\mathbb{E}_{\hat{x},\bar{x}}[\mathit{REPA}(\bar{x},\hat{x})] \le 1 &- \tfrac{1}{8}\,\mathrm{MMD}_{\mathrm{DINOv2}}\big(p_X, p_{\hat{X}}\big) \\ &+ \tfrac{1}{2}\,\mathbb{E}_{x,\bar{x}}[\mathrm{ApproxErr}(x,\bar{x})] + \tfrac{1}{4}\,\mathbb{E}_{\hat{x}}[\mathrm{MisREPA}(\hat{x})]\,.\end{aligned}$$

*where*

$$\mathrm{MMD}_{\mathrm{DINOv2}}\big(p_X,\, p_{\hat{X}}\big) = \left\| \mathbb{E}_x[\mu_f(x)] - \mathbb{E}_{\hat{x}}[\mu_f(\hat{x})] \right\|_2^2$$

*is the maximum mean discrepancy between $p_X$ and $p_{\hat{X}}$ in the DINOv2 feature space.*

[1]When the timestep argument is omitted, all $t$-dependent quantities are evaluated at $t = 0$. In particular, $\mathrm{MisREPA}(x) \equiv \mathrm{MisREPA}(x,0)$ and $\mathrm{DiffEnc}^{[n]}(x) \equiv \mathrm{DiffEnc}^{[n]}(x,0)$.

The proof is provided in the Appendix A.1. Proposition 4.3 establishes a connection between the representation alignment term and the maximum mean discrepancy (MMD) [2], which quantifies perceptual divergence between feature distributions. This relationship indicates that *maximizing* the REPA term encourages the *minimization* of a divergence measure captured by the MMD together with additional penalties accounting for feature misalignment between diffusion and DINOv2 representations, as well as an approximation error arising from the use of proxy rather than ground-truth features for alignment. When these terms are sufficiently small, Proposition 4.3 implies that maximizing the REPA term effectively corresponds to minimizing the maximum mean discrepancy (MMD) between the DINOv2 feature distributions of reconstructed and ground-truth images. This interpretation reveals that REPA regularization offers a principled, representation-based mechanism for enforcing perceptual consistency, by guiding generated images to align with the semantic structure captured by the DINOv2 encoder.

**Effect of REPA on Diffusion Representations.** We now study how a single REPA update influences the latent diffusion state relative to that of the clean image.

**Proposition 4.4** (informal)**.** *Consider an inverse problem with observation $y \in \mathbb{R}^m$ generated from a clean image $x \sim p_X$, and let $\bar{x}$ denote the proxy used by REPA. Let $z_t \in \mathbb{R}^d$ be the diffusion state at time $t$, $z_t^{(\mathrm{REPA})}$ its REPA-regularized update. Let $z^* \in \mathbb{R}^d$ denote the latent variable associated with the clean image $x$. Under regularity assumptions on the Jacobian of the diffusion encoder $J_{\mathrm{DiffEnc}}$ and the mapping $g_\phi$ (see Appendix A.2), there exists a radius $r > 0$ such that, whenever $\|z_t - z^*\|_2 < r$, the REPA update induces a contraction in the diffusion state space. In particular,*

$$\|z_t^{(\mathrm{REPA})} - z^*\|_2 \le C_1 \|z_t - z^*\|_2 + e(x),$$
$$e(x) := C_2 \left( \sqrt{\mathrm{ApproxErr}(x, \bar{x})} + \sqrt{\mathrm{MisREPA}(x)} \right).$$

*where $C_1 \in (0, 1)$ and $C_2 > 0$ are constants that depend on the conditioning of $J_{\mathrm{DiffEnc}}$ and $g_\phi$.*

The proof is provided in Appendix A.1. Proposition 4.4 characterizes how the REPA update modifies the latent diffusion state. The inequality shows that, when the current state $z_t$ is reasonably close to the clean latent code $z^*$, a single REPA step contracts their distance up to two additive terms: the proxy approximation error and the representation misalignment, both of which are expected to be small in practice. This contraction therefore holds in a local regime around $z^*$, which corresponds in practice to stages of the diffusion process where the global structure of the reconstruction is already correct and the remaining errors are primarily fine-scale details and artifacts. In this regime, REPA acts as a perceptual regularizer, refining local image details and correcting artifacts that are not effectively addressed by the baseline update alone. Qualitative examples illustrating this behavior are provided in Figure 2 and the Appendix A.19.

Intuitively, Proposition 4.3 shows that the REPA regularizer induces a reduction in the discrepancy between reconstructed and ground-truth images in the DINOv2 feature space an embedding space known to correlate well with human perceptual similarity [13]. Proposition 4.4 complements this view by establishing that the REPA update also acts directly on the latent diffusion state, reducing its distance to the clean image latent during the reverse process.

Taken together, Propositions 4.3 and 4.4 indicate that REPA promotes perceptually meaningful reconstructions by steering the model toward regions of the latent space that reflect the semantic structure of natural images. Empirical evaluations in Section 5 reinforce this interpretation across multiple solvers and inverse problems.

## 5 Experimental Results

In this section, we present experimental results demonstrating the effectiveness of our alignment regularizer across a range of inverse problems. We evaluate reconstruction quality using four widely adopted metrics: PSNR (peak signal-to-noise ratio), SSIM (structural similarity index) [42], LPIPS (learned perceptual image patch similarity) [49], and FID (Fréchet Inception Distance) [14]. Experiments are conducted on the ImageNet and FFHQ datasets [9, 18], with FFHQ images resized to $256 \times 256$. For both datasets, PSNR, SSIM, and LPIPS are averaged over 100 images from the corresponding validation splits, while FID is computed over the resulting sets of reconstructions.

We consider four inverse problems: super-resolution, box inpainting, Gaussian deblurring, and motion deblurring. Following the standard corruption operators used in state-of-the-art diffusion-




Figure 2: Qualitative comparison of inverse problem reconstruction with and without REPA. The top block corresponds to box inpainting and the bottom block to Gaussian deblurring, while the left and right panels correspond to ImageNet and FFHQ, respectively. Within each panel, the first row corresponds to Latent DPS and the second row to ReSample.

based inverse problem methods [6, 48], we adopt the same degradation models in our experiments. For super-resolution, images are downsampled by a factor of $4$. For box inpainting, we mask out a $128 \times 128$ square region. Gaussian deblurring is performed using a $61 \times 61$ kernel with standard deviation $3.0$, while motion blur is simulated using a kernel of the same size and intensity $0.5$. For ImageNet, we use additive Gaussian measurement noise with standard deviation $0.01$, while for FFHQ, we increase the noise level to $0.05$ to obtain a more challenging reconstruction setting.

We provide additional experiments in the appendix to further assess the effectiveness and applicability of our approach. In Appendix A.16, we extend the evaluation to 5,000 FFHQ images to investigate whether the improvements in perceptual reconstruction quality persist when considering a substantially larger evaluation set. We also examine the applicability of our method to more challenging measurement operators by considering two nonlinear inverse problems on ImageNet in Appendix A.8. Finally, in Appendix A.18, we investigate sparse-view CT reconstruction using a domain-specific representation encoder, demonstrating that REPA can improve perceptual reconstruction quality even when the measurements lie outside the image domain and the underlying data distribution differs substantially from natural images.

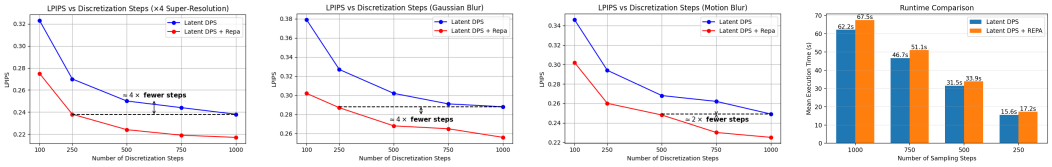
### 5.1 Effectiveness of the REPA Regularizer

This subsection evaluates how the REPA regularizer influences reconstruction quality by applying it to two representative latent-space solvers. To this end, we instantiate Algorithm 1 with Latent DPS and ReSample [37]. A detailed description of these methods and their integration with REPA is provided in Appendix A.2. For ImageNet experiments, we use the latent diffusion model trained with representation alignment from [47], together with its pretrained MLP that projects diffusion representations into the DINOv2 space. For FFHQ, we train a representation-aligned SIT-BASE model from scratch following the same setup as [47]. In ImageNet experiments, the regularizer is applied to representations extracted after the eighth transformer block, while for FFHQ it is applied after the fourth block, we include an ablation study about this choice in Appendix A.9.

Our approach can be extended to independently pretrained diffusion models by learning a mapping between the diffusion representations and the DINOv2 feature space. We study this setting in Appendix A.15. In this case, incorporating the REPA regularizer continues to provide perceptual

Table 1: Performance comparison on ImageNet ($\sigma = 0.01$) and FFHQ ($\sigma = 0.05$) across four inverse problem tasks: $4\times$ super-resolution, box inpainting, Gaussian deblurring, and motion deblurring. We evaluate the effect of incorporating REPA regularisation into Latent DPS and ReSample.

| **Task** | **Method** | **ImageNet** | | | | **FFHQ** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | LPIPS↓ | FID↓ | PSNR↑ | SSIM↑ | LPIPS↓ | FID↓ | PSNR↑ | SSIM↑ |
| $4\times$ SR | Latent DPS | 0.238 | 123.82 | 26.88 | 0.732 | 0.188 | 56.69 | 28.99 | 0.814 |
| | Latent DPS + REPA | 0.217 | 86.88 | 26.82 | 0.731 | 0.176 | 51.27 | 29.12 | 0.819 |
| | Resample | 0.208 | 97.02 | 26.70 | 0.736 | 0.178 | 52.82 | 29.06 | 0.814 |
| | Resample + REPA | **0.197** | **74.70** | 26.99 | 0.740 | **0.167** | **46.36** | **29.17** | 0.818 |
| | DPS | 0.244 | 92.13 | 24.42 | 0.681 | 0.182 | 54.19 | 27.67 | 0.804 |
| | Latent DAPS | 0.252 | 120.57 | **27.13** | **0.744** | 0.206 | 74.82 | 28.95 | **0.820** |
| Inpainting | Latent DPS | 0.151 | 116.53 | 20.53 | 0.822 | 0.192 | 65.89 | 23.55 | 0.785 |
| | Latent DPS + REPA | **0.139** | 88.69 | 20.45 | 0.824 | **0.178** | **57.04** | 23.83 | 0.784 |
| | Resample | 0.153 | 121.98 | 20.53 | 0.812 | 0.192 | 67.99 | 23.86 | 0.792 |
| | Resample + REPA | 0.143 | **87.15** | 20.79 | **0.827** | **0.178** | 61.97 | 24.01 | 0.793 |
| | DPS | 0.191 | 97.95 | 19.11 | 0.769 | 0.190 | 104.53 | 20.02 | 0.786 |
| | Latent DAPS | 0.318 | 188.44 | **21.21** | 0.718 | 0.210 | 89.21 | **24.39** | **0.817** |
| Gaussian Deblur | Latent DPS | 0.288 | 152.96 | 25.76 | 0.648 | 0.192 | 60.65 | 28.15 | 0.783 |
| | Latent DPS + REPA | 0.256 | 102.99 | 25.66 | 0.669 | 0.186 | 55.53 | 28.21 | 0.787 |
| | Resample | 0.259 | 115.47 | 26.19 | 0.699 | 0.172 | 56.41 | 27.01 | 0.753 |
| | Resample + REPA | **0.223** | **89.67** | **26.49** | **0.707** | **0.168** | **52.73** | 27.17 | 0.756 |
| | DPS | 0.366 | 157.73 | 19.55 | 0.461 | 0.177 | 53.18 | 27.19 | 0.789 |
| | Latent DAPS | 0.291 | 159.67 | 26.02 | 0.692 | 0.229 | 104.43 | **28.73** | **0.809** |
| Motion Deblur | Latent DPS | 0.249 | 129.08 | 27.19 | 0.738 | 0.170 | 52.14 | 27.20 | 0.773 |
| | Latent DPS + REPA | 0.225 | 90.23 | 27.01 | 0.735 | 0.165 | 47.18 | 27.16 | 0.772 |
| | Resample | 0.210 | 89.95 | 27.57 | 0.739 | 0.157 | 52.19 | 28.36 | 0.791 |
| | Resample + REPA | **0.192** | **75.11** | **27.80** | **0.766** | 0.151 | 50.02 | 28.41 | 0.794 |
| | DPS | 0.242 | 89.06 | 24.17 | 0.678 | **0.146** | **43.32** | 27.28 | 0.793 |
| | Latent DAPS | 0.264 | 125.18 | 27.25 | 0.744 | 0.191 | 76.21 | **29.77** | **0.836** |




(a) Super Resolution (b) Gaussian Deblurring (c) Motion Deblurring (d) Runtime Comparison

Figure 3: (a)–(c) Comparison of LPIPS as a function of sampling steps for super-resolution, Gaussian deblurring, and motion deblurring, respectively. REPA achieves comparable perceptual reconstruction quality with fewer sampling steps. (d) Comparison of execution times with and without REPA for super-resolution, showing that the additional regularisation introduces only a modest computational overhead. Importantly, the substantial reduction in sampling steps enabled by REPA translates into significantly lower execution times, as illustrated by the comparison between 250 and 1,000 sampling steps.

improvements for inverse problems. In line with our theory, the method remains effective as long as the learned mapping is sufficiently accurate, corresponding to a low MisREPA error.

Table 1 reports quantitative results across all four inverse problems. Incorporating REPA yields consistent improvements in perceptual metrics, notably reducing LPIPS and FID for both Latent DPS and ReSample while maintaining comparable PSNR and SSIM values. This behaviour is consistent with the perception–distortion trade-off [5], whereby improvements in perceptual realism do not necessarily translate into gains in pixel-wise metrics. In Appendix A.12, we provide qualitative examples illustrating that REPA can produce sharper and more perceptually realistic reconstructions even when the corresponding PSNR is lower. We further investigate this phenomenon by varying the

regularisation strength, demonstrating how the choice of alignment strength influences the balance between perceptual quality and reconstruction fidelity.

Figure 2 presents qualitative comparisons for box inpainting and Gaussian deblurring, where reconstructions obtained with REPA are visually sharper and more faithful to the ground truth. In addition to improving these latent solvers, we also compare our approach with other state-of-the-art reconstruction algorithms. For pixel-space methods, we consider DPS [6], evaluated using the pretrained diffusion model from [10]. For latent diffusion baselines, we include Latent DAPS, implemented with the conditional ImageNet latent diffusion model from [32]. As shown in Table 1, the proposed alignment regularizer consistently achieves state-of-the-art performance in perception-based metrics across the considered inverse problems.

**Effect of REPA with Varying Discretization Steps.** Figure 3 illustrates how LPIPS varies with the number of discretization steps across three inverse problems, together with the corresponding runtime comparison. Incorporating the REPA regularizer consistently reduces perceptual error compared to Latent DPS alone. More importantly, REPA enables comparable or better perceptual quality with substantially fewer sampling steps: for example, in super-resolution and Gaussian deblurring, REPA achieves the same LPIPS with roughly $4\times$ fewer steps, while in motion deblurring it achieves the same level with about $2\times$ fewer steps. The fourth plot in Figure 3 shows the runtime comparison for different numbers of discretization steps. The reported runtime is averaged over 100 images for the super resolution problem. While REPA introduces additional computation per step, the reduced number of required sampling steps allows the method to reach the same perceptual quality in less overall runtime. All runtime experiments were run on a single NVIDIA L40 GPU (48GB).

**REPA vs Guidance in the DINO feature space.** To further showcase the benefits of representation alignment, Appendix A.13 compares REPA to a variant that enforces consistency directly in the DINOv2 feature space of the reconstructed image. Concretely, the feature-only variant applies the following guidance at each timestep:

$$z_{t-1} \leftarrow z_{t-1} + \lambda \nabla_{z_t} \sum_{n=1}^{N} \cos\Big( c_{\text{proxy}}^{[n]}, f_{\text{DINOv2}}^{[n]}\big(\mathcal{D}(\mathbb{E}[z_0 \mid z_t])\big)\Big),$$

This approach relies solely on the DINOv2 features of the decoded image. In contrast, REPA aligns the model's internal representations with DINOv2 features. As observed in Appendix A.13, this feature-only guidance performs worse in practice. A natural explanation through the lens of Proposition 4.4 is that the composite mapping

$$z_t \mapsto \mathcal{D}(\mathbb{E}[z_0 \mid z_t]) \mapsto f_{\text{DINOv2}}(\cdot)$$

is significantly more ill-conditioned than the lightweight projection used in REPA, resulting in weaker and less stable gradients with respect to the diffusion state.

## 6 Conclusions

In this work, we introduced *representation alignment* for solving inverse problems with diffusion- and flow-based generative models. By aligning internal model representations with features from a pretrained DINOv2 encoder, our method provides a strong semantic prior that guides the reconstruction process toward perceptually meaningful solutions. We further offered theoretical insights into this mechanism, showing that REPA regularization is related to minimizing a divergence in the DINOv2 feature space and can reduce the discrepancy between the current and the clean latent state under suitable conditions. Our extensive experiments across different inverse problems and datasets demonstrate consistent improvements in perceptual reconstruction quality when incorporating REPA into existing inverse problem solvers. Moreover, we show that representation alignment can achieve comparable reconstruction quality with substantially fewer sampling steps, reducing the overall computational cost. Importantly, our approach can also be applied to pretrained generative models through a lightweight projection head, without requiring retraining of the underlying generative model. Overall, our findings establish representation alignment as a flexible and effective inference-time regularization strategy for improving generative inverse problem solvers, highlighting the potential of pretrained visual representations to guide reconstruction toward perceptually realistic solutions.

## References


[1] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

[2] Michael Arbel, Anna Korba, Adil Salim, and Arthur Gretton. Maximum mean discrepancy gradient flow. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.

[4] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. *Advances in Neural Information Processing Systems*, 35:15309–15324, 2022.

[5] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6228–6237, 2018.

[6] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023.

[7] Corentin Dancette, Julien Khlaut, Antoine Saporta, Helene Philippe, Elodie Ferreres, Baptiste Callard, Théo Danielou, Léo Alberge, Léo Machado, Daniel Tordjman, et al. Curia: A multi-modal foundation model for radiology. *arXiv preprint arXiv:2509.06830*, 2025.

[8] Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems. *arXiv preprint arXiv:2410.00083*, 2024.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[11] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in Neural Information Processing Systems*, 29, 2016.

[12] Julius Erbach, Dominik Narnhofer, Andreas Dombos, Bernt Schiele, Jan Eric Lenssen, and Konrad Schindler. Solving inverse problems with flair. *Advances in Neural Information Processing Systems*, 38:136667–136704, 2026.

[13] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023.

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

[16] Vincent Tao Hu, David W Zhang, Yuki M Asano, Gertjan J Burghouts, and Cees GM Snoek. Self-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18413–18422, 2023.

[17] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[19] Chieh-Hsin Lai, Yang Song, Dongjun Kim, Yuki Mitsufuji, and Stefano Ermon. The principles of diffusion models. *arXiv preprint arXiv:2510.21890*, 2025.

[20] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. *arXiv preprint arXiv:2504.10483*, 2025.

[21] Johannes Leuschner, Maximilian Schmidt, Daniel Otero Baguer, and Peter Maaß. The lodopab-ct dataset: A benchmark dataset for low-dose ct reconstruction methods. *arXiv preprint arXiv:1910.01113*, 2019.

[22] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2206–2217, 2023.

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

[24] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.

[25] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[26] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *International Conference on Learning Representations*, volume 2024, pages 28027–28053, 2024.

[27] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[28] Jean Pachebat, Giovanni Conforti, Alain Durmus, and Yazid Janati. Iterative tilting for diffusion fine-tuning. *arXiv preprint arXiv:2512.03234*, 2025.

[29] Maitreya Patel, Song Wen, Dimitris N Metaxas, and Yezhou Yang. Steering rectified flow models in the vector field for controlled image generation. *arXiv preprint arXiv:2412.00100*, 2024.

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[31] Ron Raphaeli, Sean Man, and Michael Elad. Silo: Solving inverse problems with latent operators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10570–10580, 2025.

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

[33] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[34] Shelly Sheynin, Oron Ashual, Adam Polyak, Uriel Singer, Oran Gafni, Eliya Nachmani, and Yaniv Taigman. Knn-diffusion: Image generation via large-scale retrieval. *arXiv preprint arXiv:2204.02849*, 2022.

[35] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

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

[37] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. In *The Twelfth International Conference on Learning Representations*, 2024.

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[39] Darshan Thaker, Abhishek Goyal, and René Vidal. Frequency-guided posterior sampling for diffusion-based image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12873–12882, 2025.

[40] Yuchuan Tian, Hanting Chen, Mengyu Zheng, Yuchen Liang, Chao Xu, and Yunhe Wang. U-repa: Aligning diffusion u-nets to vits. *arXiv preprint arXiv:2503.18414*, 2025.

[41] Chenyu Wang, Cai Zhou, Sharut Gupta, Zongyu Lin, Stefanie Jegelka, Stephen Bates, and Tommi Jaakkola. Learning diffusion models with flexible representation guidance. *arXiv preprint arXiv:2507.08980*, 2025.

[42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[43] Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, Xuanlei Zhao, Pengfei Zhou, Kaipeng Zhang, Zhangyang Wang, et al. Repa works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. *arXiv preprint arXiv:2505.16792*, 2025.

[44] Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[45] Wanghan Xu, Xiaoyu Yue, Zidong Wang, Yao Teng, Wenlong Zhang, Xihui Liu, Luping Zhou, Wanli Ouyang, and Lei Bai. Exploring representation-aligned latent space for better generation. *arXiv preprint arXiv:2502.00359*, 2025.

[46] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15703–15712, 2025.

[47] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

[48] Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20895–20905, 2025.

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

# A Appendix

## A.1 Proofs

**Proposition A.1.** *Assume $x \sim p_X$ is a ground-truth image and $\hat{x} \sim p_{\hat{X}|Y}$ is its corresponding reconstruction, where $p_{\hat{X}|Y}$ is the distribution induced by any reconstruction method. Assume further that all DINOv2 feature embeddings are $\ell_2$–normalized. Let $\bar{x}$ be a proxy approximation of $x$ Define the empirical mean DINOv2 feature embedding as*

$$\mu_f(x) := \mathbb{E}_n\Big[f^{[n]}_{\text{DINOv2}}(x)\Big].$$

*Then the expected REPA alignment satisfies*

$$\begin{aligned}\mathbb{E}_{\hat{x},\bar{x}}[\textit{REPA}(\bar{x},\hat{x})] \;\le\; 1 - \frac{1}{8}\,\text{MMD}_{\text{DINOv2}}\big(p_X,\, p_{\hat{X}}\big) \\ + \frac{1}{2}\,\mathbb{E}_{x,\bar{x}}\big[\text{ApproxErr}(x,\bar{x})\big] + \frac{1}{4}\,\mathbb{E}_{\hat{x}}\big[\text{MisREPA}(\hat{x})\big].\end{aligned}$$

*where*

$$\text{MMD}_{\text{DINOv2}}\big(p_X,\, p_{\hat{X}}\big) = \big\|\mathbb{E}_x[\mu_f(x)] - \mathbb{E}_{\hat{x}}[\mu_f(\hat{x})]\big\|_2^2$$

*is the maximum mean discrepancy between $p_X$ and $p_{\hat{X}}$ in the DINOv2 feature space.*

*Proof.*

$$\text{REPA}(\bar{x},\hat{x}) = \frac{1}{N}\sum_{n=1}^{N}\cos\Big(f^{[n]}_{\text{DINOv2}}(\bar{x}),\, g_\phi\big(\textsc{DiffEnc}^{[n]}(\hat{x})\big)\Big),$$

Assume the features are $\ell_2$–normalized. Then for each $n$,

$$\cos\Big(f^{[n]}_{\text{DINOv2}}(\bar{x}),\, g_\phi\big(\textsc{DiffEnc}^{[n]}(\hat{x})\big)\Big) = 1 - \tfrac{1}{2}\,\big\|f^{[n]}_{\text{DINOv2}}(\bar{x}) - g_\phi(\textsc{DiffEnc}^{[n]}(\hat{x}))\big\|_2^2,$$

and hence

$$\text{REPA}(\bar{x},\hat{x}) = 1 \;-\; \frac{1}{2N}\sum_{n=1}^{N}\big\|f^{[n]}_{\text{DINOv2}}(\bar{x}) - g_\phi(\textsc{DiffEnc}^{[n]}(\hat{x}))\big\|_2^2.$$

Let the empirical (mean) feature embeddings be

$$\mu_f(x) := \frac{1}{N}\sum_{n=1}^{N} f^{[n]}_{\text{DINOv2}}(x), \qquad \mu_f(\bar{x}) := \frac{1}{N}\sum_{n=1}^{N} f^{[n]}_{\text{DINOv2}}(\bar{x}),$$

$$\mu_f(\hat{x}) := \frac{1}{N}\sum_{n=1}^{N} f^{[n]}_{\text{DINOv2}}(\hat{x}), \qquad \mu_g(\hat{x}) := \frac{1}{N}\sum_{n=1}^{N} g_\phi(\textsc{DiffEnc}^{[n]}(\hat{x})).$$

From Jensen's inequality for the squared norm :

$$\big\|\mu_f(x) - \mu_f(\hat{x})\big\|_2^2 \le 2\,\big\|\mu_f(x) - \mu_g(\hat{x})\big\|_2^2 + 2\,\big\|\mu_f(\hat{x}) - \mu_g(\hat{x})\big\|_2^2.$$

Moreover,

$$\big\|\mu_f(x)-\mu_g(\hat{x})\big\|_2^2 = \Big\|\frac{1}{N}\sum_{n=1}^{N}\big(f^{[n]}_{\text{DINOv2}}(x)-g_\phi(\textsc{DiffEnc}^{[n]}(\hat{x}))\big)\Big\|_2^2 \;\le\; \frac{1}{N}\sum_{n=1}^{N}\big\|f^{[n]}_{\text{DINOv2}}(x)-g_\phi(\textsc{DiffEnc}^{[n]}(\hat{x}))\big\|_2^2,$$

and similarly

$$\big\|\mu_f(\hat{x}) - \mu_g(\hat{x})\big\|_2^2 \;\le\; \frac{1}{N}\sum_{n=1}^{N}\big\|f^{[n]}_{\text{DINOv2}}(\hat{x}) - g_\phi(\textsc{DiffEnc}^{[n]}(\hat{x}))\big\|_2^2 \;= \text{MisREPA}(\hat{x}).$$

Combining the two yields

$$\big\|\mu_f(x)-\mu_f(\hat{x})\big\|_2^2 \le \frac{2}{N}\sum_{n=1}^{N}\big\|f^{[n]}_{\text{DINOv2}}(x)-g_\phi(\text{DIFFENC}^{[n]}(\hat{x}))\big\|_2^2 + 2\,\text{MisREPA}(\hat{x})$$

$$\le \frac{4}{N}\sum_{n=1}^{N}\big\|f^{[n]}_{\text{DINOv2}}(\bar{x})-g_\phi(\text{DIFFENC}^{[n]}(\hat{x}))\big\|_2^2 + \frac{4}{N}\sum_{n=1}^{N}\big\|f^{[n]}_{\text{DINOv2}}(x)-f^{[n]}_{\text{DINOv2}}(\bar{x})\big\|_2^2 + 2\,\text{MisREPA}(\hat{x}). \tag{8}$$

$$= \frac{4}{N}\sum_{n=1}^{N}\big\|f^{[n]}_{\text{DINOv2}}(\bar{x})-g_\phi(\text{DIFFENC}^{[n]}(\hat{x}))\big\|_2^2 + 4\text{ApproxErr}(x,\bar{x}) + 2\text{MisREPA}(\hat{x}). \tag{9}$$

Using the cosine–squared identity for unit–norm features,

$$\sum_{n=1}^{N}\big\|f^{[n]}_{\text{DINOv2}}(\bar{x})-g_\phi(\text{DIFFENC}^{[n]}(\hat{x}))\big\|_2^2 = 2N\big(1-\text{REPA}(\bar{x},\hat{x})\big),$$

$$\big\|\mu_f(x)-\mu_f(\hat{x})\big\|_2^2 \;\le\; 8\Big(1-\text{REPA}(\bar{x},\hat{x})\Big) \;+\; 4\text{ApproxErr}(x,\bar{x}) \;+\; 2\,\text{MisREPA}(\hat{x}).$$

Rearranging gives the bound

$$\text{REPA}(\bar{x},\hat{x}) \;\le\; 1-\frac{1}{8}\big\|\mu_f(x)-\mu_f(\hat{x})\big\|_2^2 + \frac{1}{2}\,\text{ApproxErr}(x,\bar{x}) + \frac{1}{4}\,\text{MisREPA}(\hat{x}).$$

Taking expectations over $x\sim p_X$, and $\hat{x}\sim p_{\hat{X}}$, and using the convexity of the squared norm,

$$\mathbb{E}\big[\|\mu_f(x)-\mu_f(\hat{x})\|_2^2\big] \;\ge\; \big\|\mathbb{E}_x[\mu_f(x)]-\mathbb{E}_{\hat{x}}[\mu_f(\hat{x})]\big\|_2^2 = \text{MMD}_{\text{DINOv2}}(p_X,p_{\hat{X}}),$$

we obtain

$$\mathbb{E}_{\hat{x},\bar{x}}[\text{REPA}(\bar{x},\hat{x})] \;\le\; 1-\frac{1}{8}\,\text{MMD}_{\text{DINOv2}}\big(p_X,\,p_{\hat{X}}\big)+\frac{1}{2}\,\mathbb{E}_{x,\bar{x}}\big[\text{ApproxErr}(x,\bar{x})\big]+\frac{1}{4}\,\mathbb{E}_{\hat{x}}\big[\text{MisREPA}(\hat{x})\big], \tag{10}$$

**Assumption A.2** (Regularity of the diffusion encoder and REPA projection)**.** We operate under the following regularity and conditioning assumptions on the diffusion feature extractor and the REPA projection. For notational simplicity in the analysis, we introduce an operator corresponding to the vectorized diffusion encoder. Let $\widetilde{G}(\cdot) := \text{vec}(\text{DIFFENC}(\cdot,t)) : \mathbb{R}^d \to \mathbb{R}^{ND}$ denote the vectorized diffusion feature extractor at timestep $t$, and let $\widetilde{g}_\phi : \mathbb{R}^{ND} \to \mathbb{R}^{ND}$ denote the patchwise lift of the projection head $g_\phi : \mathbb{R}^D \to \mathbb{R}^D$. Fix two diffusion states $z_t, z^* \in \mathbb{R}^d$ and define $u_t := \widetilde{G}(z_t)$ and $u^* := \widetilde{G}(z^*)$. Let

$$B := \{z^* + s(z_t - z^*) : s \in [0,1]\} \quad \text{and} \quad \mathcal{U} := \{\widetilde{G}(z) : z \in B\}.$$

**A.2.1** The mapping $\widetilde{G}$ is differentiable, and its Jacobian $J_{\widetilde{G}}(z) \in \mathbb{R}^{(ND)\times d}$ is Lipschitz continuous on $B$ with Lipschitz constant $L_G$.

**A.2.2** There exist constants $m_G > 0$ and $M_G < \infty$ such that

$$J_{\widetilde{G}}(z_t)^\top J_{\widetilde{G}}(z_t) \succeq m_G^2 I_d, \qquad \|J_{\widetilde{G}}(z)\|_2 \le M_G \quad \text{for all } z \in B.$$

**A.2.3** The mapping $\widetilde{g}_\phi$ is differentiable, and its Jacobian $J_{\widetilde{g}_\phi}(u) \in \mathbb{R}^{(ND)\times(ND)}$ is uniformly well-conditioned at $u_t$: there exists a constant $m_\phi > 0$ such that

$$J_{\widetilde{g}_\phi}(u_t)^\top J_{\widetilde{g}_\phi}(u_t) \succeq m_\phi^2 I_{ND}, \qquad \|J_{\widetilde{g}_\phi}(u)\|_2 \le M_\phi \quad \text{for all } u \in \mathcal{U}.$$

**A.2.4** The Jacobian $J_{\widetilde{g}_\phi}$ is Lipschitz continuous on $\mathcal{U}$ with Lipschitz constant $L_\phi$.

**Proposition A.3** (Local contraction of REPA in the diffusion state space)**.** *Consider an inverse problem in which an observation $y \in \mathbb{R}^m$ is generated from a ground-truth image $x \sim p_X$, and let $\bar{x} \sim p_{\bar{X}}$ denote the proxy approximation used during REPA. Let $z_t \in \mathbb{R}^d$ be the diffusion state at timestep $t$ produced by Algorithm 1, and let $z_t^{(\mathrm{REPA})}$ denote the diffusion state after applying the REPA regularization update. Let $z^* \in \mathbb{R}^d$ denote the latent variable associated with the clean image $x$. Under Assumptions $A_1$–$A_4$, define*

$$r := \frac{m_G^2\, m_\phi^2}{L_\phi M_G^3 M_\phi + L_G M_\phi^2 M_G.}$$

*If $\|z_t - z^*\|_2 < r$, then there exists a stepsize*

$$0 < \lambda \leq \frac{1}{2M_G^2\, M_\phi^2}$$

*such that the REPA update induces a local contraction in the diffusion state space. In particular,*

$$\|z_t^{(\mathrm{REPA})} - z^*\|_2 \;\leq\; C_1 \|z_t - z^*\|_2 + C_2 \left(\sqrt{\mathrm{ApproxErr}(x, \bar{x})} + \sqrt{\mathrm{MisREPA}(x)}\right),$$

*where*

$$C_1 := 1 - \lambda m_G^2 m_\phi^2 \in (0, 1), \qquad C_2 := 2\,\lambda\, M_G\, M_\phi \sqrt{N}.$$

*Proof.* We first rewrite all quantities in vectorized form. Let $\mathrm{vec}(\cdot)$ denote the operator that stacks the rows of a matrix into a single vector. For any matrix $H \in \mathbb{R}^{k_1 \times k_2}$, define

$$\widetilde{H} := \mathrm{vec}(H) \in \mathbb{R}^{k_1 k_2}.$$

For the DINOv2 features $f_{\mathrm{DINOv2}}(x) \in \mathbb{R}^{N \times D}$ we write

$$\widetilde{f}(x) := \mathrm{vec}\big(f_{\mathrm{DINOv2}}(x)\big) \in \mathbb{R}^{ND}.$$

For the diffusion-model feature extractor $\textsc{DiffEnc} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^{N \times D}$, we define its vectorized form by

$$\widetilde{G}(z_t) := \mathrm{vec}\big(\textsc{DiffEnc}(z_t, t)\big) \in \mathbb{R}^{ND}.$$

Recall that $\phi : \mathbb{R}^D \to \mathbb{R}^D$ is differentiable. Its lifted Jacobian acts patchwise and satisfies

$$J_{\widetilde{g}_\phi}(\widetilde{G}(z_t)) = I_N \otimes J_\phi(G_2(z_t)).$$

The REPA update is given by

$$z_t^{(\mathrm{REPA})} = z_t - \lambda \nabla_{z_t} \big\|\widetilde{f}(\bar{x}) - \widetilde{g}_\phi(\widetilde{G}(z_t))\big\|_2^2.$$

By the chain rule,

$$z_t^{(\mathrm{REPA})} = z_t + 2\lambda\, J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top \big(\widetilde{f}(\bar{x}) - \widetilde{g}_\phi(\widetilde{G}(z_t))\big).$$

Subtracting $z^*$ from both sides yields

$$z_t^{(\mathrm{REPA})} - z^* = z_t - z^* + 2\lambda\, J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top \big(\widetilde{f}(\bar{x}) - \widetilde{g}_\phi(\widetilde{G}(z_t))\big).$$

We now add and subtract $\widetilde{f}(x)$ and $\widetilde{g}_\phi(\widetilde{G}(z^*))$ inside the parentheses:

$$\begin{aligned} z_t^{(\mathrm{REPA})} - z^* = (z_t - z^*) &- 2\lambda\, J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top \big(\widetilde{g}_\phi(\widetilde{G}(z_t)) - \widetilde{g}_\phi(\widetilde{G}(z^*))\big) \\ &+ 2\lambda\, J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top \big(\widetilde{f}(\bar{x}) - \widetilde{f}(x)\big) \\ &+ 2\lambda\, J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top \big(\widetilde{f}(x) - \widetilde{g}_\phi(\widetilde{G}(z^*))\big). \end{aligned} \tag{11}$$

We define the *contractive part* as

$$\mathcal{T}_t := (z_t - z^*) - 2\lambda\, J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top \big(\widetilde{g}_\phi(\widetilde{G}(z_t)) - \widetilde{g}_\phi(\widetilde{G}(z^*))\big).$$

Applying the fundamental theorem of calculus to $u \mapsto \widetilde{g}_\phi(u)$ along $u(s) = \widetilde{G}(z(s))$ with $z(s) = z^* + s(z_t - z^*)$, $s \in [0,1]$, yields

$$\widetilde{g}_\phi(\widetilde{G}(z_t)) - \widetilde{g}_\phi(\widetilde{G}(z^*)) = \int_0^1 J_{\widetilde{g}_\phi}\big(\widetilde{G}(z(s))\big)\, J_{\widetilde{G}}\big(z(s)\big)\, (z_t - z^*)\, ds.$$

Substituting into the definition of $\mathcal{T}_t$ gives

$$\mathcal{T}_t = \Big[I - 2\lambda \int_0^1 J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}\big(\widetilde{G}(z_t)\big)^\top J_{\widetilde{g}_\phi}\big(\widetilde{G}(z(s))\big) J_{\widetilde{G}}\big(z(s)\big)\, ds\Big](z_t - z^*). \tag{12}$$

Add and subtract $J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\, J_{\widetilde{G}}(z_t)$ inside the integral to obtain

$$\begin{aligned}\mathcal{T}_t = &\Big[I - 2\lambda\, J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t)) J_{\widetilde{G}}(z_t)\Big](z_t - z^*) \\ &- 2\lambda\Big[\int_0^1 J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top \Big(J_{\widetilde{g}_\phi}(\widetilde{G}(z(s)))\, J_{\widetilde{G}}(z(s)) - J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\, J_{\widetilde{G}}(z_t)\Big)\, ds\Big](z_t - z^*).\end{aligned} \tag{13}$$

Taking Euclidean norms in (13) and using the triangle inequality gives

$$\begin{aligned}\|\mathcal{T}_t\|_2 \le &\Big\|I - 2\lambda\, J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t)) J_{\widetilde{G}}(z_t)\Big\|_2\, \|z_t - z^*\|_2 \\ &+ 2\lambda\Big\|\int_0^1 J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top \Big(J_{\widetilde{g}_\phi}(\widetilde{G}(z(s)))\, J_{\widetilde{G}}(z(s)) - J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\, J_{\widetilde{G}}(z_t)\Big)\, ds\Big\|_2\, \|z_t - z^*\|_2.\end{aligned} \tag{14}$$

Moreover, by submultiplicativity of the spectral norm and $\big\|\int_0^1 A(s)\, ds\big\|_2 \le \int_0^1 \|A(s)\|_2\, ds$, the second term can be bounded as

$$\Big\| \int_0^1 J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top \Big(J_{\widetilde{g}_\phi}(\widetilde{G}(z(s)))\, J_{\widetilde{G}}(z(s)) - J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\, J_{\widetilde{G}}(z_t)\Big)\, ds\Big\|_2 \le \tag{15}$$

$$\int_0^1 \|J_{\widetilde{G}}(z_t)\|_2\, \|J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\|_2 \Big\|J_{\widetilde{g}_\phi}(\widetilde{G}(z(s)))\, J_{\widetilde{G}}(z(s)) - J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\, J_{\widetilde{G}}(z_t)\Big\|_2\, ds. \tag{16}$$

Combining (14)–(16) yields

$$\begin{aligned}\|\mathcal{T}_t\|_2 \le &\Big\|I - 2\lambda\, J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t)) J_{\widetilde{G}}(z_t)\Big\|_2\, \|z_t - z^*\|_2 \\ &+ 2\lambda\Big(\int_0^1 \|J_{\widetilde{G}}(z_t)\|_2\, \|J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\|_2 \Big\|J_{\widetilde{g}_\phi}(\widetilde{G}(z(s)))\, J_{\widetilde{G}}(z(s)) - J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\, J_{\widetilde{G}}(z_t)\Big\|_2\, ds\Big)\|z_t - z^*\|_2.\end{aligned} \tag{17}$$

We bound the difference inside the integral by writing

$$\begin{aligned}&\Big\|J_{\widetilde{g}_\phi}(\widetilde{G}(z(s)))\, J_{\widetilde{G}}(z(s)) - J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\, J_{\widetilde{G}}(z_t)\Big\|_2 \\ &\le \Big\|\big(J_{\widetilde{g}_\phi}(\widetilde{G}(z(s))) - J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\big) J_{\widetilde{G}}(z(s))\Big\|_2 + \Big\|J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\big(J_{\widetilde{G}}(z(s)) - J_{\widetilde{G}}(z_t)\big)\Big\|_2 \\ &\le \|J_{\widetilde{g}_\phi}(\widetilde{G}(z(s))) - J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\|_2\, \|J_{\widetilde{G}}(z(s))\|_2 + \|J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\|_2\, \|J_{\widetilde{G}}(z(s)) - J_{\widetilde{G}}(z_t)\|_2.\end{aligned} \tag{18}$$

Using the Lipschitz property of $J_{\widetilde{G}}$,

$$\|J_{\widetilde{G}}(z(s)) - J_{\widetilde{G}}(z_t)\|_2 \le L_G\, \|z(s) - z_t\|_2 = L_G(1-s)\, \|z_t - z^*\|_2,$$

together with the Lipschitz continuity of $J_{\widetilde{g}_\phi}$,

$$\|J_{\widetilde{g}_\phi}(\widetilde{G}(z(s))) - J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\|_2 \le L_\phi M_G (1-s)\, \|z_t - z^*\|_2,$$

and the uniform bounds

$$\|J_{\widetilde{G}}(z(s))\|_2 \le M_G, \qquad \|J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\|_2 \le M_\phi,$$

This yields

$$\Big\|J_{\widetilde{g}_\phi}(\widetilde{G}(z(s)))\,J_{\widetilde{G}}(z(s)) - J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\,J_{\widetilde{G}}(z_t)\Big\|_2 \le (1-s)\Big(L_\phi M_G^2 + L_G M_\phi\Big)\|z_t - z^*\|_2.$$

Substituting this bound into (17) and integrating over $s \in [0,1]$ yields

$$\begin{aligned}\|\mathcal{T}_t\|_2 \le \Big\|I - 2\lambda\, J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t)) J_{\widetilde{G}}(z_t)\Big\|_2 \,\|z_t - z^*\|_2 \\ + \lambda\Big(L_\phi M_G^3 M_\phi + L_G M_\phi^2 M_G\Big)\|z_t - z^*\|_2^2. \end{aligned} \tag{19}$$

Since the symmetric matrix

$$S_t := J_{\widetilde{G}}(z_t)^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))^\top J_{\widetilde{g}_\phi}(\widetilde{G}(z_t)) J_{\widetilde{G}}(z_t)$$

has eigenvalues contained in $[m_G^2 m_\phi^2,\ M_G^2 M_\phi^2]$, its spectral norm satisfies

$$\big\|I - 2\lambda S_t\big\|_2 = \max_{\sigma \in [m_G^2 m_\phi^2,\, M_G^2 M_\phi^2]} |1 - 2\lambda\sigma|.$$

For any $0 < \lambda \le 1/(2M_G^2 M_\phi^2)$, the maximum is attained at $\sigma = m_G^2 m_\phi^2$, and hence

$$\big\|I - 2\lambda S_t\big\|_2 \le 1 - 2\lambda m_G^2 m_\phi^2.$$

Substituting this bound into (19) yields

$$\|\mathcal{T}_t\|_2 \le \Big(1 - 2\lambda m_G^2 m_\phi^2 + \lambda\Big(L_\phi M_G^3 M_\phi + L_G M_\phi^2 M_G\Big)\|z_t - z^*\|_2\Big)\|z_t - z^*\|_2. \tag{20}$$

Therefore, if

$$\|z_t - z^*\|_2 < \frac{m_G^2\, m_\phi^2}{L_\phi M_G^3 M_\phi + L_G M_\phi^2 M_G} \qquad \text{and} \qquad 0 < \lambda \le \frac{1}{2M_G^2 M_\phi^2},$$

then the contractive part satisfies the strict contraction

$$\|\mathcal{T}_t\|_2 \le \big(1 - \lambda m_G^2 m_\phi^2\big)\,\|z_t - z^*\|_2,$$

with contraction factor strictly smaller than one. By the definitions of the approximation and misalignment errors,

$$\|\widetilde{f}(\bar{x}) - \widetilde{f}(x)\|_2 = \sqrt{N}\,\sqrt{\mathrm{ApproxErr}(x,\bar{x})}, \qquad \|\widetilde{f}(x) - \widetilde{g}_\phi\,(\widetilde{G}(z^*))\|_2 = \sqrt{N}\,\sqrt{\mathrm{MisREPA}(x)}.$$

Returning to the decomposition we combine the contraction bound for $\mathcal{T}_t$ with the remaining two terms. Using the triangle inequality together with the uniform bounds $\|J_{\widetilde{G}}(z_t)\|_2 \le M_G$ and $\|J_{\widetilde{g}_\phi}(\widetilde{G}(z_t))\|_2 \le M_\phi$, we obtain

$$\begin{aligned}\|z_t^{(\mathrm{repa})} - z^*\|_2 \le \|\mathcal{T}_t\|_2 + 2\lambda M_G M_\phi\,\|\widetilde{f}(\bar{x}) - \widetilde{f}(x)\|_2 \\ + 2\lambda M_G M_\phi\,\|\widetilde{f}(x) - \widetilde{g}_\phi(\,\widetilde{G}(z^*))\|_2. \end{aligned} \tag{21}$$

Substituting the contraction estimate together with the above identities yields

$$\|z_t^{(\mathrm{REPA})} - z^*\|_2 \le (1 - \lambda m_G^2 m_\phi^2)\,\|z_t - z^*\|_2 + C\Big(\sqrt{\mathrm{ApproxErr}(x,\bar{x})} + \sqrt{\mathrm{MisREPA}(x)}\Big), \tag{22}$$

where the constant $C$ is defined explicitly as

$$C := 2\lambda M_G M_\phi \sqrt{N}.$$

This concludes the proof.

**Algorithm 2** Latent DPS + REPA Algorithm

**Require:** flow model $u_\theta$, measurement $y$, pretrained encoder $f$, proxy representation $c_{\text{proxy}}$, timestep $t_{\text{cutoff}}$, initialize $z_T \sim \mathcal{N}(0, I)$

1: **for** $t \in \{T, \dots, 0\}$ **do**
2: $\quad v \leftarrow u_\theta(z_t, t)$
3: $\quad \hat{z}_0 \leftarrow \mathbb{E}[z_0 \mid z_t]$
4: $\quad z_{t-1} \leftarrow z_t - \frac{1}{T} \cdot v$
5: $\quad z_{t-1} \leftarrow z_{t-1} - \eta \nabla_{z_t} \|y - \mathcal{A}(\mathcal{D}(\hat{z}_0))\|_2^2$
6: $\quad z_{t-1} \leftarrow z_{t-1} + \lambda \nabla_{z_t} \sum_{n=1}^{N} \cos(c_{\text{proxy}}^{[n]}, g_\phi(\text{DiffEnc}^{[n]}(z_t, t))$
7: $\quad$ **if** $t < t_{\text{cutoff}}$ **then**
8: $\quad\quad c_{\text{proxy}} \leftarrow f_{\text{DINOv2}}\big(\mathcal{D}(\mathbb{E}[z_0 \mid z_t])\big)$
9: $\quad$ **end if**
10: **end for**
11: **return** $x_0$

### A.2 Details About Latent DPS + REPA Implementation

We implement Latent DPS + REPA following Algorithm 2 and use 1000 sampling steps throughout. We employ an SNR-based learning rate schedule of the form

$$\eta(t) = \frac{\kappa}{\max\left(\frac{t}{1-t}, 1\right)},$$

where $\kappa$ is a tunable scaling factor. This schedule accounts for the varying signal-to-noise ratio across diffusion timesteps and provides stable performance in practice. The parameters $\kappa$ and the representation-alignment strength $\lambda$ are tuned on a small validation set with present the best parameters found in Table 2.

Table 2: Hyperparameters used for Latent DPS + REPA.

| | **ImageNet** | | **FFHQ** | |
|---|---|---|---|---|
| **Task** | $\boldsymbol{\kappa}$ | $\boldsymbol{\lambda}$ | $\boldsymbol{\kappa}$ | $\boldsymbol{\lambda}$ |
| Super-resolution | 2.0 | 0.01 | *1.25* | *0.025* |
| Gaussian deblurring | 0.25 | 0.05 | *0.5* | *0.05* |
| Motion deblurring | 0.5 | 0.01 | *0.5* | *0.05* |
| Box inpainting | 0.5 | 0.01 | *0.75* | *0.025* |

### A.3 Details About Resample + REPA Implementation

In Table 3, we present an adaptation of the algorithm proposed in [37] to the flow-based setting, augmented with the REPA regularizer as described in the methodology section. We note that the original ReSample algorithm employs a three-stage procedure for enforcing data consistency: (i) gradient steps in latent space, similar to Latent DPS; (ii) pixel-space optimization, which is computationally efficient and captures high-level semantics but often leads to blurrier reconstructions; and (iii) latent space optimization as outlined to line 7 of Algorithm 3. In contrast, our adaptation omits the pixel-space stage, as our focus is on maximizing perceptual quality. We found that this modification together with the inclusion of the REPA reguralizer yields sharper and more visually convincing results, while still benefiting from the refinement effect of the final latent-space consistency updates. The best hyperparameters used for this variant are reported in Table 3. In addition to the parameters $\kappa$ and $\lambda$, which play analogous roles to those in Latent DPS, we also tune the maximum number of inner-loop optimization steps and the corresponding inner learning rate of the Resample procedure (see Algorithm 3).

Table 3: Hyperparameters used for Resampling + REPA.

| | ImageNet | | | | FFHQ | | | |
|---|---|---|---|---|---|---|---|---|
| **Task** | $\kappa$ | $\lambda$ | **max iters** | **inner lr** | $\kappa$ | $\lambda$ | **max iters** | **inner lr** |
| Super-resolution | 0.05 | 3.25 | 150 | 0.005 | 0.01 | 2.25 | 100 | 0.0001 |
| Gaussian deblurring | 0.075 | 0.5 | 300 | 0.005 | 0.05 | 0.75 | 300 | 0.00075 |
| Motion deblurring | 0.5 | 0.75 | 300 | 0.005 | 0.05 | 0.75 | 100 | 0.005 |
| Box inpainting | 0.025 | 100 | 200 | 0.0005 | 0.05 | 0.75 | 200 | 0.0005 |

**Algorithm 3** Resample + REPA Algorithm

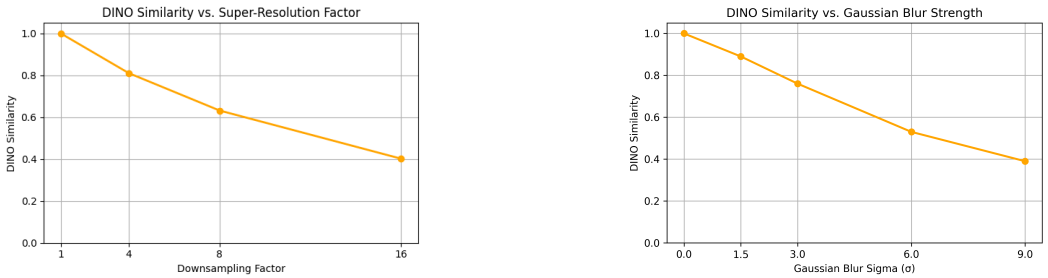
**Require:** flow model $u_\theta$, measurement $y$, pretrained encoder $f$, resample steps $C$, parameter $\gamma$, proxy representation $c_{\text{proxy}}$, timestep $t_{\text{cutoff}}$, initialize $z_T \sim \mathcal{N}(0, I)$

**for** $t \in \{T, \dots, 0\}$ **do**
  $\hat{z}_0 \leftarrow \mathbb{E}[z_0 \mid z_t]$
  $z_{t-1} \leftarrow z_t - \frac{1}{T} \cdot u_\theta(z_t, t)$
  **if** $t \in C$ **then**
    $\tilde{z}_0(y) \leftarrow \arg\min_z \frac{1}{2}\|y - \mathcal{A}(\mathcal{D}(z))\|_2^2$ ▷ initialize at $\hat{z}_0$
    $z_{t-1} \leftarrow \text{STOCHASTICRESAMPLE}(\tilde{z}_0(y),\, z_{t-1},\, \gamma)$
  **end if**
  $z_{t-1} \leftarrow z_{t-1} + \lambda \nabla_{z_t} \sum_{n=1}^{N} \cos\big(c_{\text{proxy}}^{[n]},\, g_\phi(\text{DiffEnc}^{[n]}(z_t, t))\big)$
  **if** $t < t_{\text{cutoff}}$ **then**
    $c_{\text{proxy}} \leftarrow f_{\text{DINOv2}}\big(\mathcal{D}(\mathbb{E}[z_0 \mid z_t])\big)$
  **end if**
**end for**
$x_0 \leftarrow \mathcal{D}(z_0)$
**return** $x_0$

### A.4 Robustness of DINOv2 Representations to Corruptions

To evaluate the robustness of DINOv2 representations under various corruptions, we conduct experiments on a fixed set of 100 images on the ImageNet dataset (Similar results hold for the FFHQ dataset as illustrated in Appendix A.6). For each image, we compute the average patch similarity between its ground truth representation and that of its corrupted version. We assess this similarity across different corruption types specifically super-resolution and Gaussian deblurring at varying levels of severity. We report the average similarity across the selected subset of 100 images.

Figure 5 first visualizes how the pixel-space appearance of a single image changes across different corruption levels. Figure 4 then shows how the DINOv2 representation similarity changes as the corruption severity increases for the two tasks. Interestingly, we observe that DINOv2 representations remain significantly more robust to both super-resolution and Gaussian deblurring. Despite substantial visual degradation in pixel space, DINOv2 features maintain a strong alignment with the original image features.




(a) Super-Resolution (b) Gaussian Deblurring

Figure 4: Similarity of representations under increasing levels of corruption.



(a) Ground Truth (b) Super Resolution ×4 (c) Super Resolution ×8 (d) Super Resolution ×16



(e) Ground Truth (f) Gaussian deblur $\sigma = 3$ (g) Gaussian deblur $\sigma = 6$ (h) Gaussian deblur $\sigma = 9$

Figure 5: Visual comparison of a corrupted images for different corruption types and severity levels.

### A.5 Selection of Proxy Representation

Since pretrained DINOv2 features are robust to a wide range of common image corruptions, we use the DINOv2 encoding of the measurement as an initial proxy representation, $f_{\text{DINOv2}}(y)$. We find this choice to be effective across all considered inverse problems. In addition, we introduce a cutoff timestep at which the proxy representation is updated to depend on the model's current denoised estimate, namely $f_{\text{DINOv2}}(\mathcal{D}(\mathbb{E}[z_0 \mid z_t]))$.

- **Box Inpainting.** We construct the proxy representation by combining the observed regions of the measurement with the model's current denoised estimate:
$$c_{\text{proxy}} = \text{mask} \odot f_{\text{DINOv2}}(y) + (I - \text{mask}) \odot f_{\text{DINOv2}}\big(\mathcal{D}(\mathbb{E}[z_0 \mid z_t])\big), \tag{23}$$
where $\text{mask}$ denotes the binary inpainting mask.
- **Super-resolution, Gaussian Deblurring, and Motion Deblurring.** For these tasks, we use $f_{\text{DINOv2}}(y)$ as the proxy representation during the early stages of sampling. On the FFHQ dataset, we find it beneficial to switch to the reconstruction-based proxy $f_{\text{DINOv2}}(\mathcal{D}(\mathbb{E}[z_0 \mid z_t]))$ after $80\%$ of the diffusion steps. On ImageNet, this transition does not lead to consistent improvements, and we therefore use the measurement-based proxy throughout. We include an ablation to justify these choices in A.10.

### A.6 Results Using a CLIP Model

In this subsection, we compare the results obtained using a DINOv2 encoder with those obtained using a CLIP model. To this end, we train a model with representation alignment using CLIP representations instead of DINOv2 representations on the FFHQ dataset. Both models were trained for 100K iterations using a batch size of 256. Table 4 presents quantitative results when applying our inference-time framework to both models. As we can see, the inclusion of our regularizer leads to consistent improvements across all tasks and metrics.

To compare the influence of DINOv2 and CLIP, we make the following observations. First, the baseline algorithm performs slightly better when the model has been trained using DINOv2 representations rather than CLIP. This result is consistent with the findings of [47], where the best-performing model was also trained using DINOv2 representations, suggesting that DINOv2-based representation alignment leads to a stronger generative prior. Regarding the comparison of our test-time regularizer between DINOv2 and CLIP, we observe that DINOv2 representations appear to be more robust to the standard degradations used in inverse problem evaluation. We illustrate this behavior in Figure 6.

Table 4: Performance comparison on FFHQ ($\sigma = 0.05$) for models trained with DINOv2 and CLIP representations.

| Task | Method | LPIPS↓ | FID↓ | PSNR↑ | SSIM↑ |
|---|---|---|---|---|---|
| $4\times$ SR | Latent DPS (DINOv2-trained) | 0.188 | 56.69 | 28.99 | 0.814 |
| | Latent DPS + REPA (DINOv2-trained) | **0.176** | **51.27** | **29.12** | **0.819** |
| | Latent DPS (CLIP-trained) | 0.190 | 63.63 | 28.95 | 0.811 |
| | Latent DPS + REPA (CLIP-trained) | 0.183 | 54.88 | 28.99 | 0.815 |
| Box Inpainting | Latent DPS (DINOv2 Trained) | 0.192 | 65.89 | 23.55 | **0.785** |
| | Latent DPS + REPA (DINOv2 Trained) | **0.178** | **57.04** | **23.83** | 0.784 |
| | Latent DPS (CLIP-trained) | 0.193 | 54.01 | 23.53 | 0.784 |
| | Latent DPS + REPA (CLIP-trained) | 0.185 | 52.45 | 23.65 | 0.784 |
| Gaussian Deblur | Latent DPS (DINOv2-trained) | 0.192 | 60.65 | 28.15 | 0.783 |
| | Latent DPS + REPA (DINOv2-trained) | **0.186** | **55.53** | **28.21** | **0.787** |
| | Latent DPS (CLIP-trained) | 0.193 | 60.72 | 28.02 | 0.782 |
| | Latent DPS + REPA (CLIP-trained) | 0.188 | 59.76 | 28.07 | 0.779 |

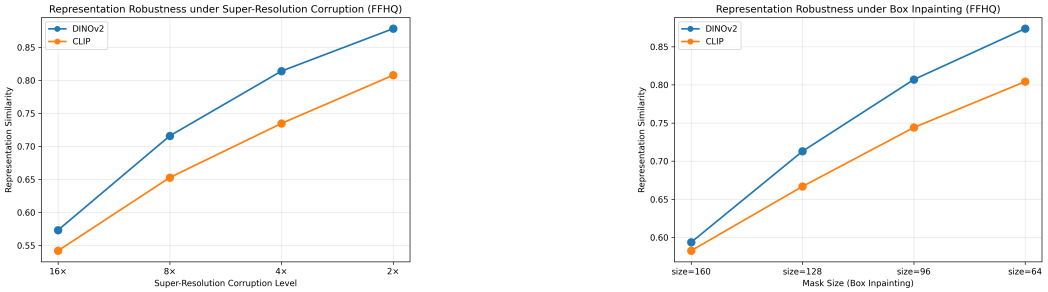
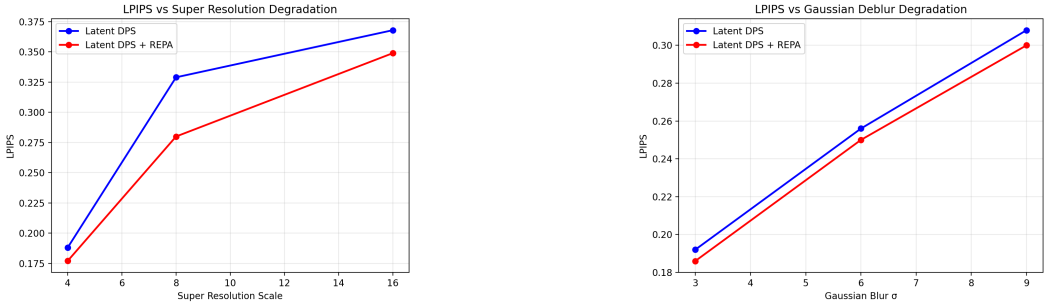



Figure 6: Comparison of the DINOv2 and CLIP robustness to common degradations.

### A.7 Solving Inverse Problems under Severe Degradations

In this subsection, we study the effect of the REPA regularizer under severe degradations. We evaluate the performance of our method using Latent DPS on the FFHQ dataset for both super-resolution and Gaussian deblurring. For super-resolution, we consider more challenging settings with $8\times$ and $16\times$ upscaling factors, while for Gaussian deblurring we consider stronger blur levels with $\sigma = 6$ and $\sigma = 9$. Figure 7 shows how LPIPS varies as the degradation severity increases. We also include the full quantitative results in Table 5.




LPIPS vs Super Resolution Degradation LPIPS vs Gaussian Deblur Degradation

Figure 7: LPIPS as a function of degradation severity for super-resolution (left) and Gaussian deblurring (right). REPA consistently improves perceptual quality across all degradation levels.

Table 5: Performance under severe degradations on FFHQ using Latent DPS.

| Task | Method | LPIPS↓ | FID↓ | PSNR↑ | SSIM↑ |
|---|---|---|---|---|---|
| 8× Super Resolution | Latent DPS | 0.329 | 92.18 | 22.85 | 0.593 |
| | Latent DPS + REPA | 0.280 | 72.70 | 23.24 | 0.617 |
| 16× Super Resolution | Latent DPS | 0.368 | 109.87 | 20.11 | 0.499 |
| | Latent DPS + REPA | 0.349 | 95.87 | 20.41 | 0.523 |
| Gaussian Deblur ($\sigma = 6$) | Latent DPS | 0.256 | 77.82 | 24.35 | 0.658 |
| | Latent DPS + REPA | 0.250 | 72.43 | 24.42 | 0.665 |
| Gaussian Deblur ($\sigma = 9$) | Latent DPS | 0.308 | 86.58 | 21.82 | 0.566 |
| | Latent DPS + REPA | 0.300 | 81.84 | 21.97 | 0.582 |

### A.8 Extension to Non Linear Problems

In this section, we extend our evaluation of the REPA regularizer to more complex inverse problems. Specifically, we evaluate our method on non-linear deblurring and high dynamic range reconstruction using the degradation model of [48]. Table 6 presents quantitative results for Latent DPS and Resample, indicating improved performance when incorporating the REPA regularizer.

Table 6: Performance comparison on ImageNet for non-linear degradation tasks.

| | | ImageNet | | | |
|---|---|---|---|---|---|
| **Task** | **Method** | LPIPS↓ | FID↓ | PSNR↑ | SSIM↑ |
| Non-linear Deblurring | Latent DPS | 0.302 | 170.41 | 24.62 | 0.653 |
| | Latent DPS + REPA | 0.266 | 105.73 | 24.52 | 0.645 |
| | Resample | 0.276 | 141.44 | 25.04 | 0.661 |
| | Resample + REPA | **0.248** | **103.96** | **25.13** | **0.668** |
| HDR Reconstruction | Latent DPS | 0.310 | 138.78 | 23.12 | 0.690 |
| | Latent DPS + REPA | **0.258** | **78.24** | **24.11** | 0.701 |
| | Resample | 0.295 | 125.00 | 23.61 | 0.701 |
| | Resample + REPA | 0.282 | 108.41 | 24.01 | **0.711** |

### A.9 Effect of the Choice of Transformer Block for Representation Alignment

We include an ablation study investigating the transformer block at which representation alignment is applied. We use the ImageNet model from [47] and study the effect of changing the alignment layer for the task of 4× super-resolution. The mapping $g_\phi$ for the ImageNet model was trained using representations from the eighth transformer block. Figure 8 shows how performance varies when alignment is applied at different transformer blocks. The reported LPIPS values are averaged over 50 images. We observe that applying alignment at different blocks still leads to performance improvements; however, the largest improvements are obtained when alignment is applied at the same block on which the mapping $g_\phi$ was trained.

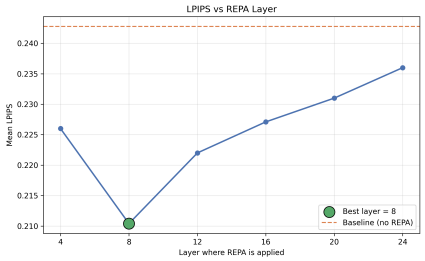



Figure 8: Ablation study on the transformer block used for representation alignment.

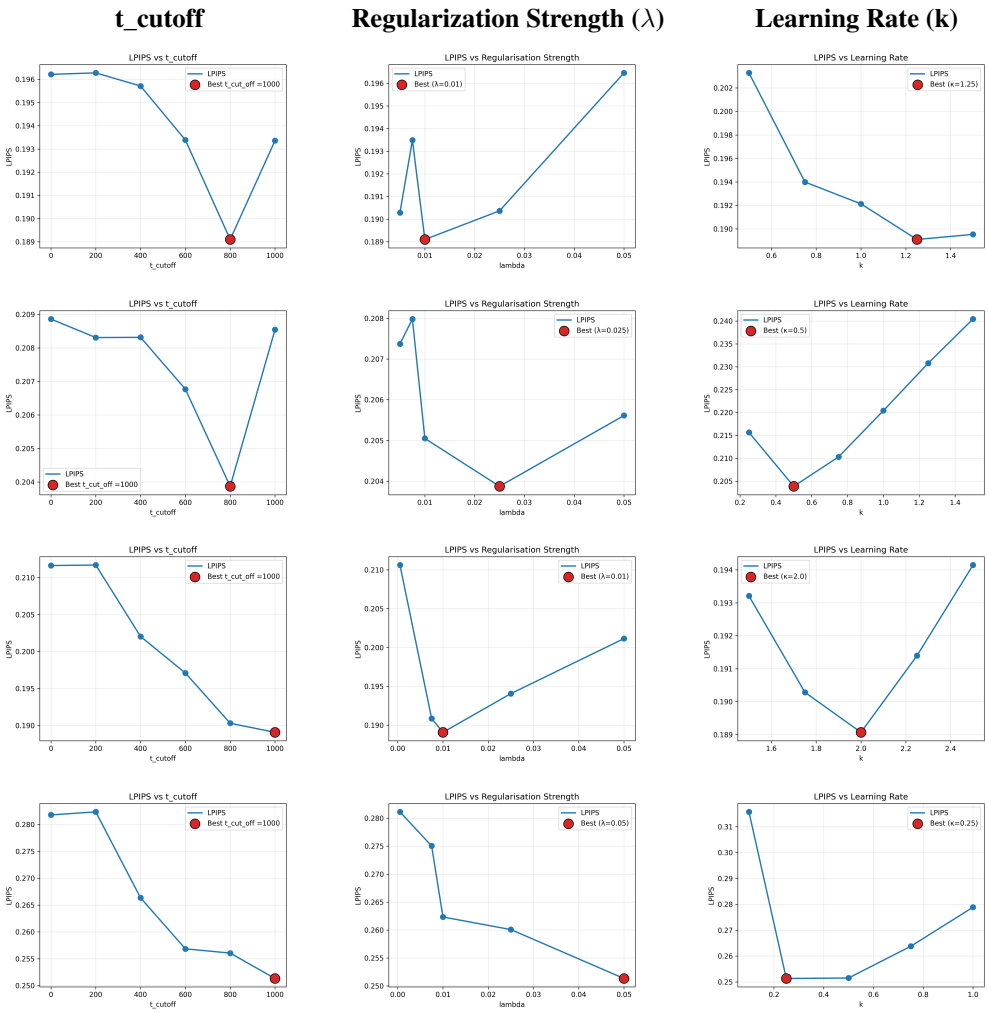



Figure 9: Ablation studies on the main hyperparameters of our framework. Rows correspond to different tasks and datasets: the first row shows FFHQ super-resolution, the second row FFHQ Gaussian deblurring, the third row ImageNet super-resolution, and the fourth row ImageNet Gaussian deblurring. Each plot shows the best LPIPS value achievable when the corresponding hyperparameter is fixed and the remaining hyperparameters are varied.

### A.10 Ablation Studies

In this subsection, we present ablation studies on the main hyperparameters of our framework. The hyperparameters were selected through a small grid search on a validation set of 15 images. The results are summarized in Figure 9. Each row corresponds to a different task and dataset, while the columns correspond to the hyperparameters $t_{\text{cutoff}}$, the regularization strength $\lambda$, and the learning rate $k$. For each hyperparameter, we report the best LPIPS value obtained when fixing that parameter and varying the remaining hyperparameters.

Regarding the selection of the cutoff timestep, we observe that switching to the denoised estimate at very early timesteps leads to significantly worse performance compared to switching later in the diffusion process. This can be explained by the fact that the denoised estimate is still very noisy during the early diffusion steps. Our framework leverages the robustness of the DINOv2 features and therefore benefits from using the denoised estimate only after a sufficiently large cutoff timestep. Interestingly, for ImageNet we found that the best performance is obtained without switching, indicating that the optimal cutoff strategy depends on the dataset and reconstruction difficulty.

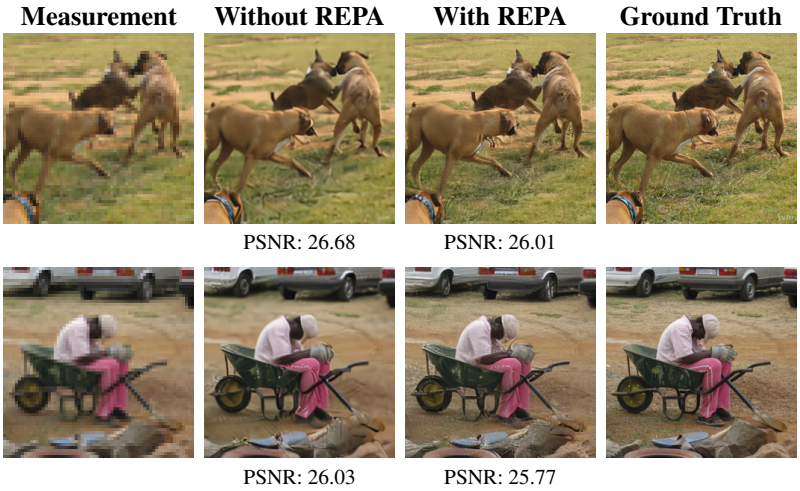



Figure 10: Qualitative comparison for $4\times$ super-resolution on the ImageNet dataset. While REPA produces sharper and more perceptually realistic reconstructions, these do not necessarily correspond to higher PSNR values, illustrating the perception–distortion trade-off.

### A.11 Experimental Details

We implement the inverse problem operators following the setup of [6]. For deblurring tasks, we follow [48] and fix a single realization of the degradation operator to ensure fair and consistent comparisons across methods. All baseline results are obtained using the official implementations provided by [6] and [48].

For DPS, we use the dataset-specific hyperparameters reported in the original paper. For DAPS, we adopt the hyperparameters provided in the official codebase for latent diffusion models.

For PSNR and SSIM, we report results obtained by averaging the reconstructed images across five independent runs of the solver and then computing the metrics on the averaged reconstruction.

### A.12 Perception Distortion Tradeoff

We include qualitative examples for the task of $4\times$ super-resolution on the ImageNet dataset. In these examples, we observe that although the reconstructions obtained with REPA are perceptually more realistic and contain sharper details, the PSNR values do not necessarily increase. More generally, we observe that PSNR often favours smoother images, whereas perceptual metrics and visual inspection favour images with sharper details.

We also investigate the effect of the regularisation strength $\lambda$ on perceptual quality and reconstruction fidelity for $4\times$ super-resolution on Imagenet. All results are calculated by averaging over 25 images. As shown in Fig. 11, increasing the regularisation strength initially improves LPIPS while maintaining nearly unchanged PSNR. As the alignment strength increases further, perceptual quality continues to improve at the expense of pixel-wise fidelity, illustrating the well-known perception–distortion trade-off [5]. These results suggest that REPA allows us to control the balance between perceptual quality and reconstruction fidelity through the choice of regularisation strength.

### A.13 Enforcing Feature-Level DINOv2 Consistency without Representation Alignment

In this ablation study, we compare REPA against a variant that enforces consistency directly in the DINOv2 feature-space, without involving the diffusion model representations. More specifically, instead of using the representation alignment loss in line 8 of Algorithm 2, we replace it with the following update step:

$$z_{t-1} \leftarrow z_{t-1} + \lambda \nabla_{z_t} \sum_{n=1}^{N} \cos\Big(c_{\text{proxy}}^{[n]},\, f_{\text{DINOv2}}^{[n]}\big(\mathcal{D}(\mathbb{E}[z_0 \mid z_t])\big)\Big)\,, \tag{24}$$

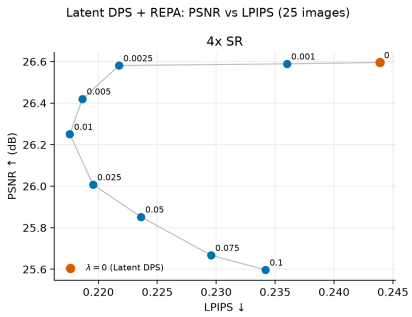



Figure 11: Perception–distortion trade-off for $4\times$ super-resolution as the REPA regularisation strength $\lambda$ varies. Lower LPIPS indicates better perceptual quality, while higher PSNR indicates better reconstruction fidelity.

Note that in this case we can only use $f_{\text{DINOv2}}(y)$ as the proxy term. While both objectives influence the sampling trajectory, the feature-only variant derives its guidance from the DINOv2 features of the decoded image. In contrast, REPA aligns the diffusion model's internal representations with DINOv2 features directly at each timestep. As shown in Table 7, the direct feature-space alignment yields consistently lower perceptual quality than the proposed representation-based formulation. for both experiments we have chosen the best hyperparameters using a small grid search as in A.2 to ensure fairness in comparison.

Table 7: Quantitative comparison between REPA and direct feature-space alignment.

| | **Super-resolution** ($\times 4$) | | **Gaussian deblurring** | |
|---|---|---|---|---|
| **Method** | **LPIPS** $\downarrow$ | **FID** $\downarrow$ | **LPIPS** $\downarrow$ | **FID** $\downarrow$ |
| Feature-space alignment | 0.227 | 95.69 | 0.276 | 128.29 |
| Representation alignment (proposed) | 0.217 | 86.88 | 0.256 | 102.99 |

### A.14 Representation Alignment on an Independently Pretrained Model

The effectiveness of REPA for inverse problems relies on the availability of a mapping $g_\phi$ between the internal representations of the diffusion model and the DINOv2 feature space. Such a mapping arises naturally when the diffusion model is pretrained with representation alignment. However, when starting from an independently pretrained diffusion model, this mapping is not readily available. We describe here how to obtain such a mapping in this setting.

A natural approach is to learn $g_\phi$ from pairs of diffusion and DINOv2 representations extracted from the same images. Specifically, given a dataset $\{x_i\}_{i=1}^{n_0} \subset \mathbb{R}^k, \quad n_0 \in \mathbb{N}$ and a pretrained encoder $f : \mathbb{R}^k \to \mathbb{R}^{N\times D_1}$, we train $g_\phi$ by maximizing the REPA objective:

$$\mathcal{L}_{\text{REPA}}(\phi) = \mathbb{E}_{x,\,t,\,\epsilon}\left[\frac{1}{N}\sum_{n=1}^{N}\cos\Big(f^{[n]}(x),\, g_\phi^{[n]}(h_t)\Big)\right], \tag{25}$$

where the intermediate diffusion representation is given by $h_t = \text{DIFFENC}(x_t, t) \in \mathbb{R}^{N\times D_2}$. Here, $f^{[n]}(x) \in \mathbb{R}^{D_1}$ denotes the corresponding patch-level representation produced by the DINOv2 encoder. Importantly, only the mapping $g_\phi$ is optimized, while both the diffusion model (through DIFFENC) and the encoder $f$ remain frozen. As a result, learning $g_\phi$ is computationally lightweight compared to full diffusion model training.

We consider two parameterizations of $g_\phi$. The first follows [47] and applies a token-wise MLP independently to each patch. The second parameterizes $g_\phi$ as a lightweight Transformer-style network operating on patch-level representations. Given input tokens $h_t \in \mathbb{R}^{N\times D_2}$, the model first applies a linear projection to an intermediate dimension $D$. The resulting tokens are augmented with learned positional embeddings and processed by a stack of $L$ Transformer encoder blocks. Each

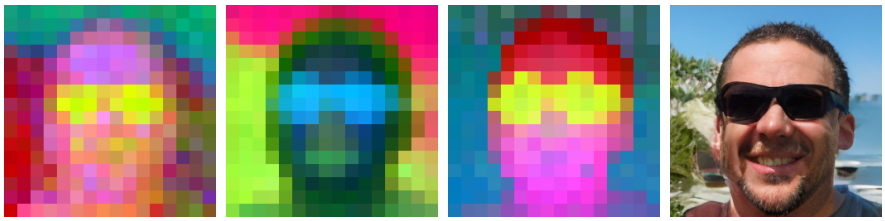


Figure 12: PCA visualizations of patch-level representations. From left to right: representations from a standard diffusion model, representations from a diffusion model trained with REPA, DINOv2 features, and the corresponding input image.

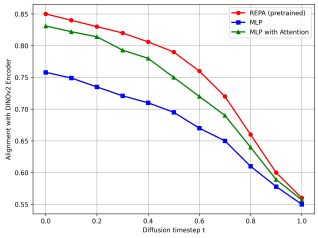



Figure 13: Alignment with the DINOv2 encoder across diffusion timesteps for different parameterizations of $g_\phi$. The red curve corresponds to a model pretrained with representation alignment, the blue curve to a token-wise MLP, and the green curve to the attention-based mapping.

block consists of a multi-head self-attention layer followed by a feed-forward network, with residual connections. Finally, a linear projection maps the output tokens to the DINOv2 feature space, yielding $g_\phi(h_t) \in \mathbb{R}^{N \times D_1}$. In our experiments, we use $D = 768$, $L = 2$, and 4 attention heads.

We train the mapping $g_\phi$ on the FFHQ dataset for 15k iterations using the alignment objective described above. Training is performed with batch size 256 and learning rate $10^{-4}$ using the AdamW optimizer. We present the resulting alignment curves in for different timesteps in Figure 13. The corresponding training setup is summarized in Table 8. We observe that the token-wise MLP is able to learn a meaningful mapping, but achieves lower alignment compared to both the attention-based parameterization and a model pretrained with representation alignment.

To motivate the need for a more expressive mapping between independently pretrained diffusion representations and the DINOv2 feature space, we visualize PCA feature maps similar to those in [47]. As shown in Figure 12, representations extracted from a standard diffusion model appear significantly noisier and less spatially coherent than those produced by a REPA-pretrained model or DINOv2. The observed structure suggests that recovering semantically aligned features benefits from aggregating information across spatial locations.

### A.15 Results with an Independently Pretrained Model

We apply our inference-time regularizer to inverse problems using the learned mapping. Table 9 reports the corresponding results, alongside those obtained with a model pretrained with representation alignment. Across all inverse problems, incorporating our regularizer consistently improves perceptual metrics (LPIPS and FID), demonstrating that learning the mapping $g_\phi$ alone is sufficient to enable effective use of REPA at inference time. More specifically, both learned mappings improve over the baseline Latent DPS (SiT), with the attention-based parameterization consistently outperforming the token-wise MLP across all tasks. While the MLP already yields gains, incorporating attention further improves LPIPS and FID. These trends are consistent with our theoretical analysis: performance is governed by the accuracy of the learned mapping, as captured by the MisREPA error. In this view, the attention-based mapping achieves lower MisREPA, leading to improved alignment and downstream performance, while models pretrained with representation alignment correspond to the smallest mismatch and therefore achieve the strongest overall results.

Table 8: Comparison of training cost across components.

| Model | Trainable parameters | Training steps | Batch size | Optimizer |
|---|---|---|---|---|
| SiT full model | 136.9M | 100k | 256 | AdamW ($10^{-4}$) |
| MLP mapping $g_\phi$ | 7.3M | 15k | 256 | AdamW ($10^{-4}$) |
| Attention mapping $g_\phi$ | 15.6M | 15k | 256 | AdamW ($10^{-4}$) |

Table 9: Performance comparison on FFHQ ($\sigma = 0.05$).

| **Task** | **Method** | **LPIPS↓** | **FID↓** | **PSNR↑** | **SSIM↑** |
|---|---|---|---|---|---|
| $4\times$ SR | Latent DPS (REPA-backbone) | 0.188 | 56.69 | 28.99 | 0.814 |
| | ↪ + REPA regularization | **0.176** | **51.27** | **29.12** | **0.819** |
| | Latent DPS (SiT backbone) | 0.190 | 61.36 | 28.92 | 0.805 |
| | ↪ + REPA (MLP mapping) | 0.184 | 56.17 | 28.95 | 0.809 |
| | ↪ + REPA (Attention mapping) | **0.177** | **54.21** | **28.97** | **0.814** |
| Gaussian Deblur | Latent DPS (REPA-backbone) | 0.192 | 60.65 | 28.15 | 0.783 |
| | ↪ + REPA regularization | **0.186** | **55.53** | **28.21** | **0.787** |
| | Latent DPS (SiT backbone) | 0.193 | 60.72 | 28.02 | 0.782 |
| | ↪ + REPA (MLP mapping) | 0.191 | 58.92 | 28.09 | 0.770 |
| | ↪ + REPA (Attention mapping) | **0.187** | **58.56** | **28.17** | **0.778** |
| Motion Deblur | Latent DPS (REPA-backbone) | 0.170 | 52.14 | **27.20** | **0.773** |
| | ↪ + REPA regularization | **0.165** | **47.18** | 27.16 | 0.772 |
| | Latent DPS (SiT backbone) | 0.171 | 54.01 | **28.02** | **0.779** |
| | ↪ + REPA (MLP mapping) | 0.171 | 52.93 | 27.73 | 0.769 |
| | ↪ + REPA (Attention mapping) | **0.167** | **49.94** | 27.89 | 0.775 |

### A.16 Large-Scale Evaluation on FFHQ

To further assess the robustness of our experimental findings and mitigate the potential bias associated with computing FID on small sample sets, we conduct an additional large-scale evaluation on 5,000 images from the FFHQ dataset. We consider two inverse problems, namely $4\times$ super-resolution and Gaussian deblurring, and compare Latent DPS with and without REPA regularisation.

Table 10: Large-scale evaluation on 5,000 FFHQ images. We compare Latent DPS with and without REPA regularisation on super-resolution and Gaussian deblurring.

| **Task** | **Method** | LPIPS↓ | FID↓ | PSNR↑ | SSIM↑ |
|---|---|---|---|---|---|
| $4\times$ SR | Latent DPS | 0.182 | 17.273 | 27.636 | 0.745 |
| | Latent DPS + REPA | **0.173** | **15.368** | **27.887** | **0.751** |
| Gaussian Deblur | Latent DPS | 0.186 | 17.168 | **27.832** | 0.739 |
| | Latent DPS + REPA | **0.179** | **16.384** | 27.827 | **0.740** |

As shown in Table 10, incorporating REPA consistently improves perceptual reconstruction quality across both inverse problems, as reflected in lower LPIPS and FID scores. Meanwhile, PSNR and SSIM remain largely comparable to those of the baseline. These results demonstrate that the perceptual improvements obtained through representation alignment persist when the evaluation is extended to a substantially larger number of images.

### A.17 Comparison with Variational Inverse-Problem Solvers

We further compare our approach against two variational diffusion-based inverse-problem solvers, RED-diff [26] and FLAIR [12]. We consider $4\times$ super-resolution and Gaussian deblurring on

Table 11: Comparison with variational inverse-problem solvers on ImageNet and FFHQ. Best results within each task and dataset are highlighted in bold.

| **Task / Dataset** | **Method** | LPIPS↓ | PSNR↑ | SSIM↑ |
|---|---|---|---|---|
| 4× SR / FFHQ | FLAIR | 0.217 | 28.35 | 0.793 |
| | RED-diff | 0.201 | **29.24** | **0.828** |
| | Latent DPS + REPA | 0.176 | 29.12 | 0.819 |
| | Resample + REPA | **0.167** | 29.17 | 0.818 |
| 4× SR / ImageNet | FLAIR | 0.247 | 25.52 | 0.701 |
| | RED-diff | 0.256 | 25.95 | 0.710 |
| | Latent DPS + REPA | 0.217 | 26.82 | 0.731 |
| | Resample + REPA | **0.197** | **26.99** | **0.740** |
| Gaussian Deblur / FFHQ | FLAIR | 0.273 | 27.38 | 0.701 |
| | RED-diff | 0.254 | 27.80 | 0.785 |
| | Latent DPS + REPA | 0.186 | **28.21** | **0.787** |
| | Resample + REPA | **0.168** | 27.17 | 0.756 |
| Gaussian Deblur / ImageNet | FLAIR | 0.296 | 24.437 | 0.647 |
| | RED-diff | 0.364 | 23.90 | 0.603 |
| | Latent DPS + REPA | 0.256 | 25.66 | 0.660 |
| | Resample + REPA | **0.223** | **26.49** | **0.707** |

ImageNet and FFHQ. For both baselines, we use the official publicly available implementations provided by the authors and tune their algorithm-specific hyperparameters under our evaluation setting.

As shown in Table 11, our REPA-regularised methods consistently outperform both RED-diff and FLAIR in terms of LPIPS across all considered tasks and datasets.

### A.18 Medical Imaging Inverse Problems

To investigate the applicability of our approach beyond natural-image inverse problems, we conduct additional experiments on sparse-view computed tomography (CT) reconstruction. Medical images exhibit substantially different characteristics from the natural images used to pretrain DINOv2.

**Experimental Setup.** We conduct experiments using the publicly available LoDoPaB-CT dataset [21]. We train an SiT model on the ground-truth CT images using the same architecture as in our FFHQ experiments. During training, we apply representation alignment using Curia [7], a medical imaging foundation model trained using the DINOv2 self-supervised learning framework on large-scale CT and MRI data. We consider sparse-view CT reconstruction following the experimental setup of ReSample [37]. For each ground-truth image, we simulate a sinogram using a parallel-beam Radon transform with 25 uniformly distributed projection angles and add Gaussian measurement noise.

**Proxy Representation Construction.** Since the measurements are acquired in the sinogram domain, they cannot be directly processed by the representation encoder. Instead, we first obtain an initial reconstruction using Latent DPS and encode it using Curia to construct the proxy representation. We subsequently use this representation to regularise a second reconstruction obtained through Latent DPS. This procedure allows REPA to operate in settings where the measurements and the reconstructed signals belong to different domains.

**Results.** We evaluate both Latent DPS and Latent DPS with REPA on 100 images from the LoDoPaB-CT validation set. The quantitative results are reported in Table 12.

Incorporating REPA improves perceptual reconstruction quality, as reflected in lower LPIPS and FID scores. We also observe a slight improvement in SSIM, while PSNR remains nearly unchanged. These results suggest that representation alignment can improve perceptual reconstruction quality in medical imaging inverse problems, even when the measurements are not directly available in the image domain.

Table 12: Sparse-view CT reconstruction results on the LoDoPaB-CT validation set using 25 projection angles.

| **Method** | LPIPS↓ | FID↓ | PSNR↑ | SSIM↑ |
|---|---|---|---|---|
| Latent DPS | 0.277 | 89.14 | **29.676** | 0.787 |
| Latent DPS + REPA | **0.271** | **83.10** | 29.659 | **0.789** |

### A.19 Additional Qualitative Results

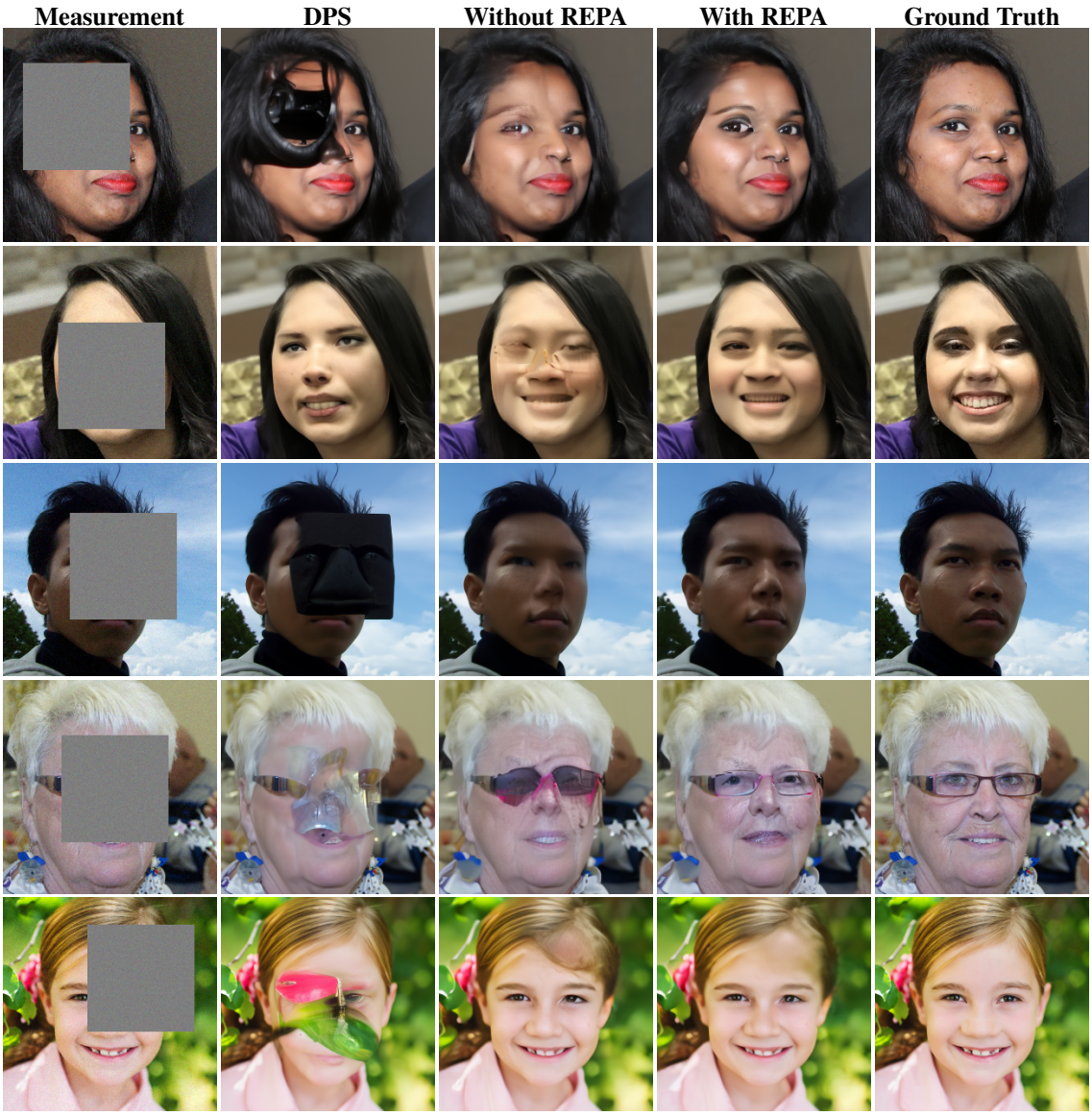



Figure 14: Qualitative comparison for box inpainting on the FFHQ dataset. Each row shows, from left to right: the measurement, DPS, the baseline latent solver, its REPA-enhanced variant, and the ground truth.

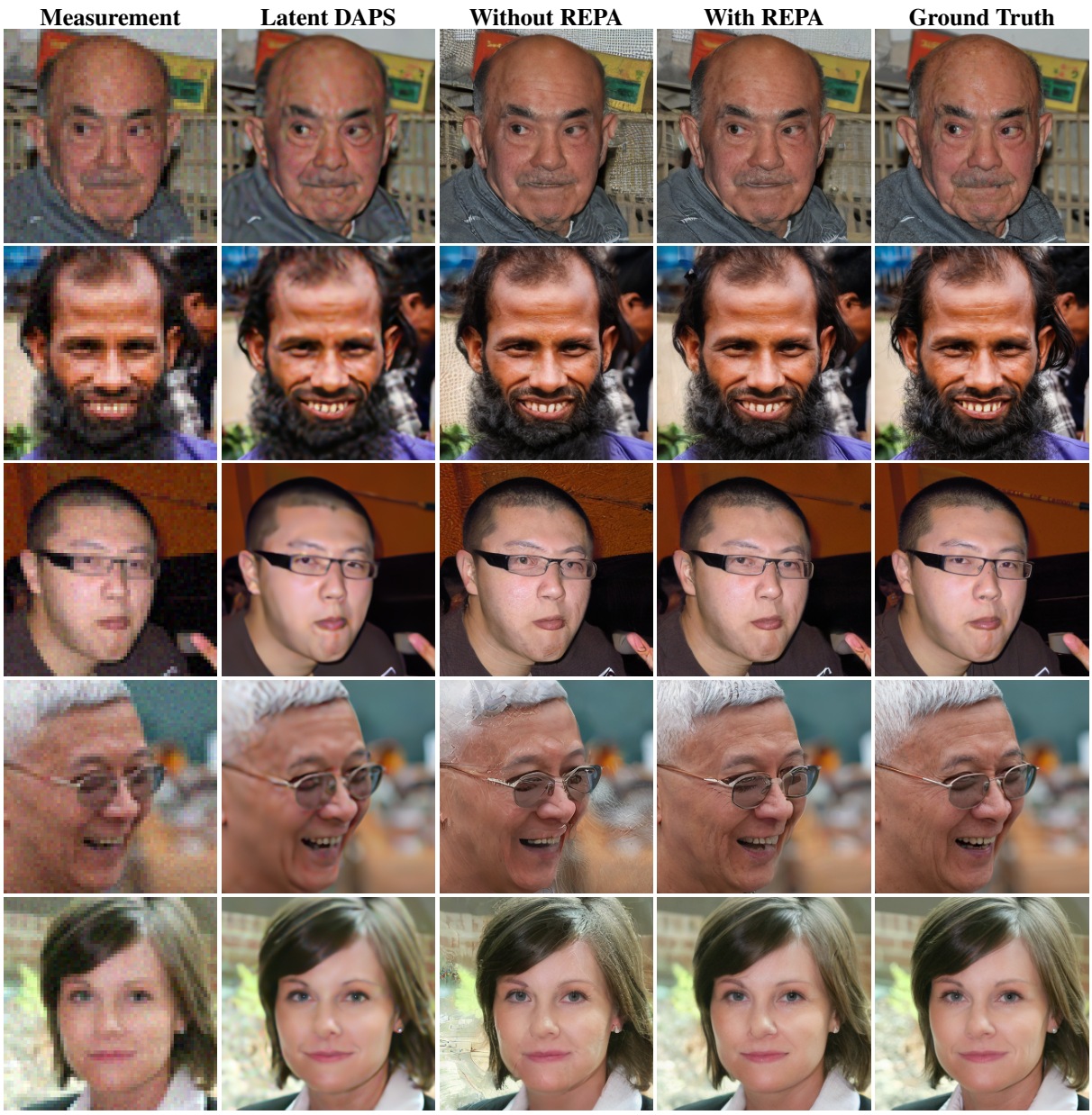



Figure 15: Qualitative comparison for $4\times$ super-resolution on the FFHQ dataset. Each row shows (from left to right): the measurement, the baseline method (Latent DAPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first two rows correspond to ReSample, while the last three correspond to Latent DPS.

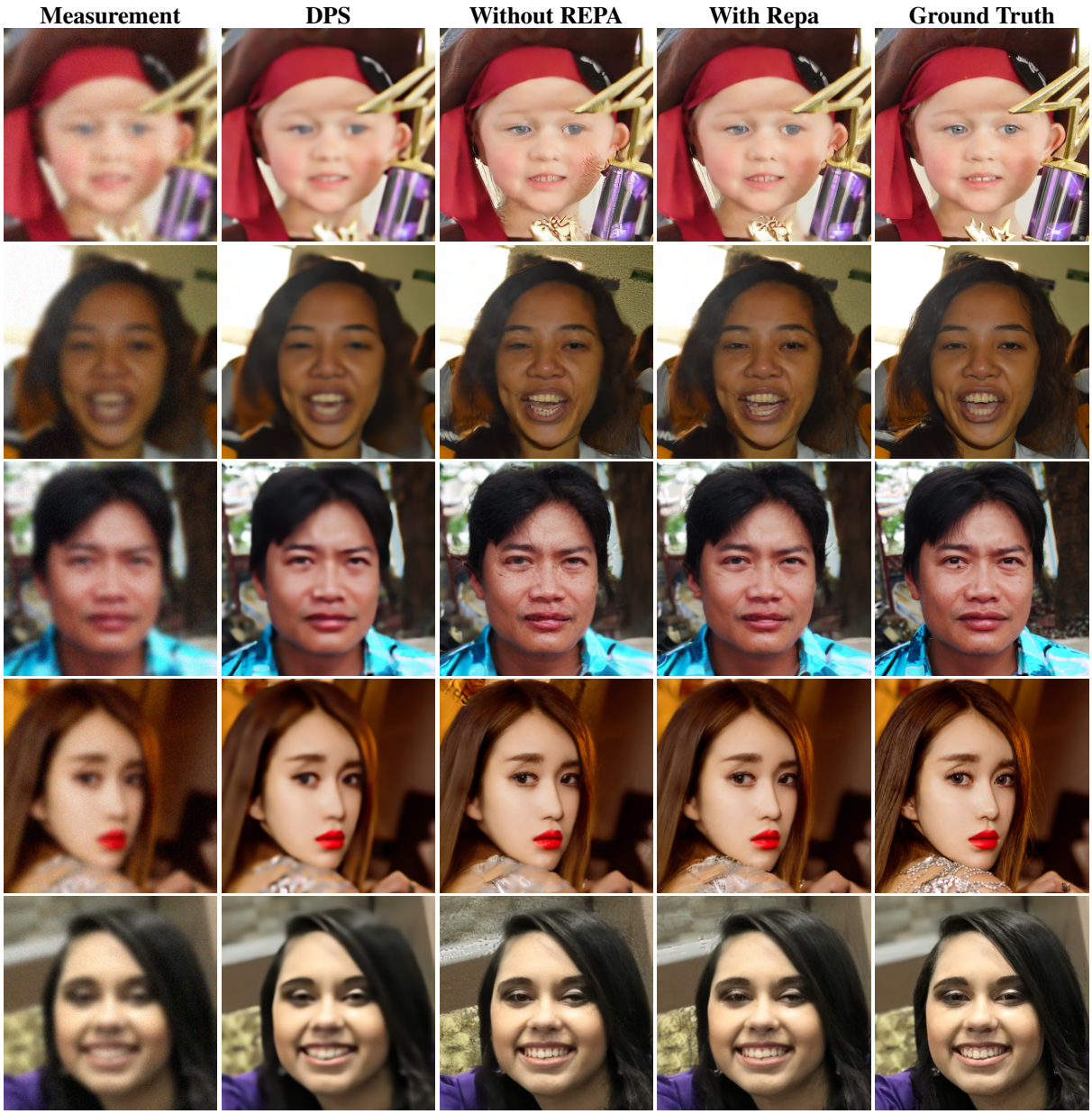



Figure 16: Qualitative comparison for Gaussian Deblurring on the FFHQ dataset. Each row shows (from left to right): the measurement, the baseline method (Latent DAPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first two rows correspond to ReSample, while the last three correspond to Latent DPS.




Figure 17: Qualitative comparison for Motion Deblurring on the FFHQ dataset. Each row shows (from left to right): the measurement, the baseline method (DPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first two rows correspond to ReSample, while the last three correspond to Latent DPS.




Figure 18: Qualitative comparison for box inpainting on the ImageNet dataset. Each row shows (from left to right): the measurement, the baseline method (DPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first three rows correspond to ReSample, while the last two correspond to Latent DPS.

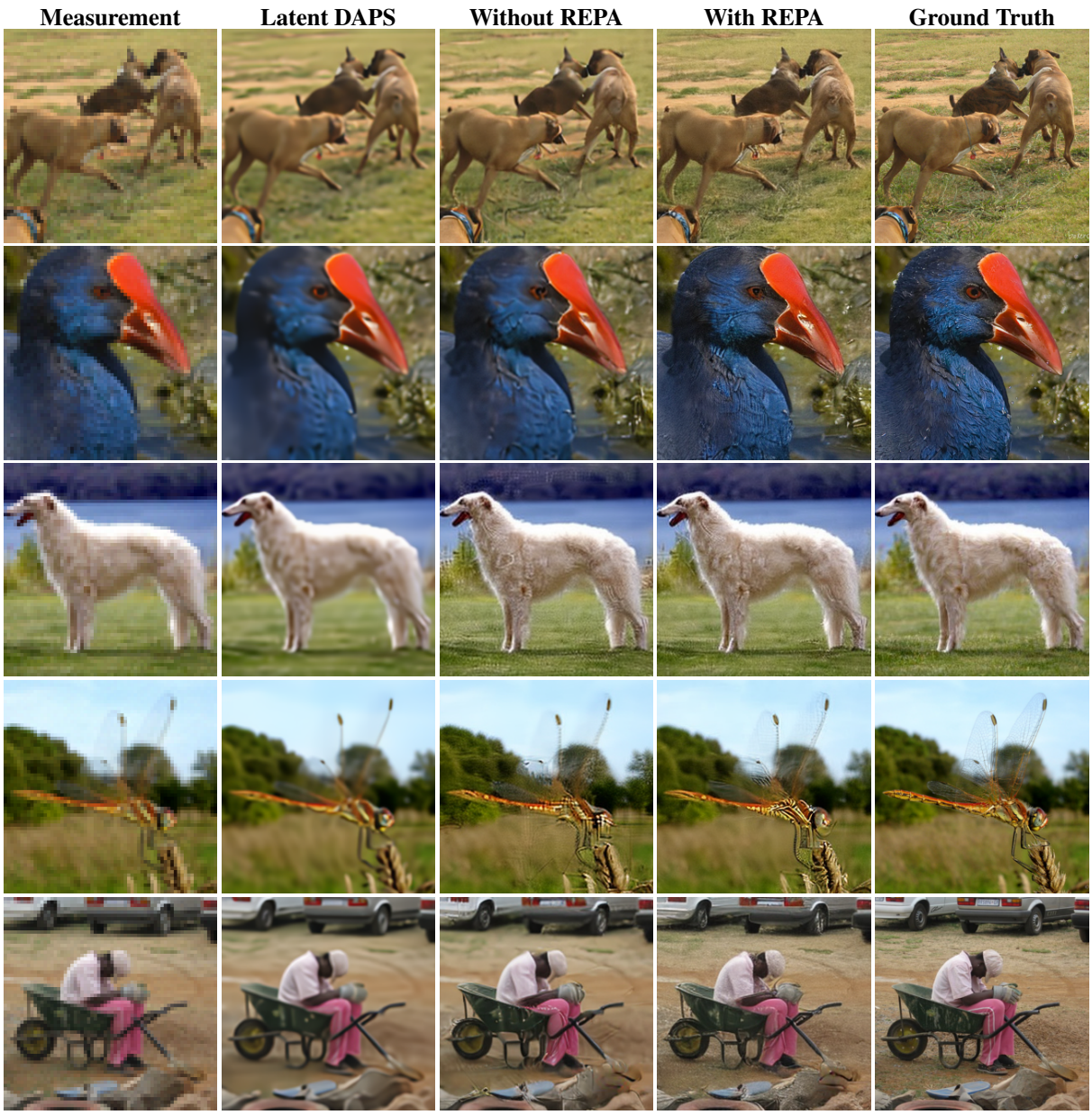



Figure 19: Qualitative comparison for Super resolution on the ImageNet dataset. Each row shows (from left to right): the measurement, the baseline method (Latent DAPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first three rows correspond to ReSample, while the last two correspond to Latent DPS.

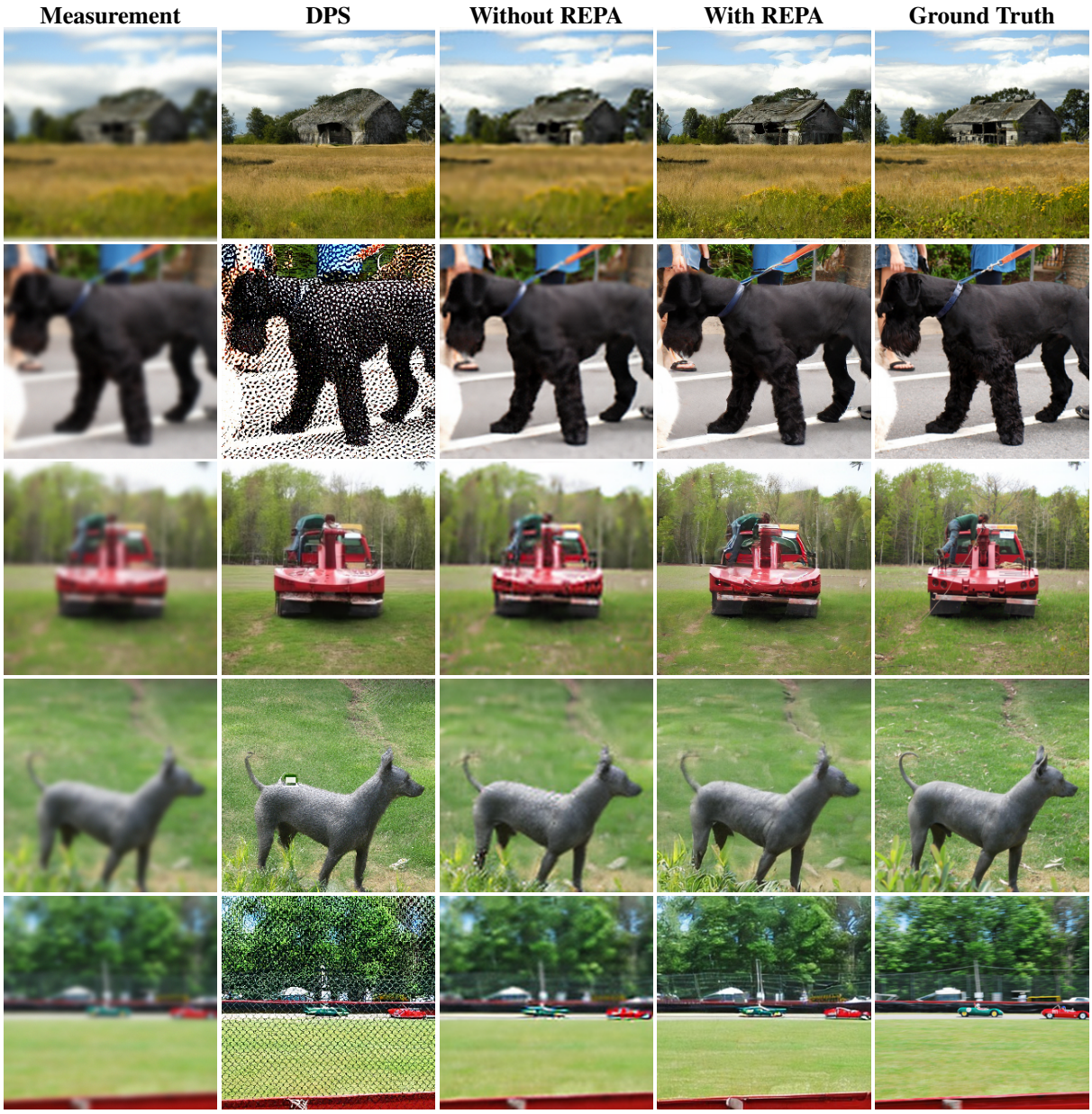



Figure 20: Qualitative comparison for Gaussian Deblurring on the ImageNet dataset. Each row shows (from left to right): the measurement, the baseline method (DPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first three rows correspond to ReSample, while the last two correspond to Latent DPS.

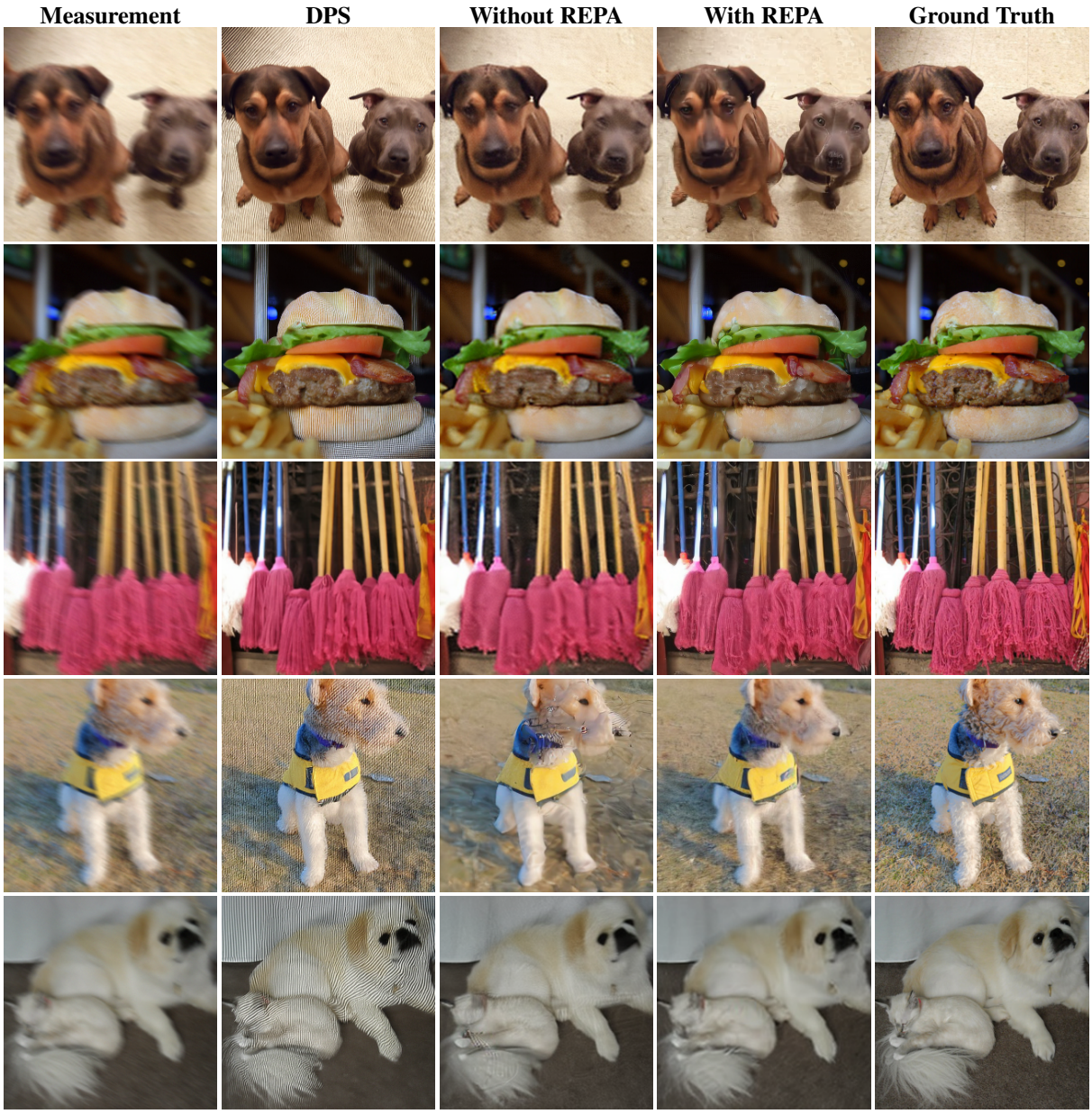



Figure 21: Qualitative comparison for Motion Deblurring on the ImageNet dataset. Each row shows (from left to right): the measurement, the baseline method (DPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first three rows correspond to ReSample, while the last two correspond to Latent DPS.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: see contributions in the introduction section.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our approach in the paper, including its reliance on the representation mapping and the quality of the proxy signal.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All assumptions are stated in the appendix for both propositions.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All relevant information can be found in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release the code and instructions to reproduce the main experimental results upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: All details can be found in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We report standard evaluation metrics on established benchmarks; however, we do not include error bars or formal statistical significance analyses.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We have included the machine under which experiments were performed in the experimental section.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research complis with the code of ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: This paper presents work aimed at advancing the field of machine learning, with a focus on improving the reliability of inverse problem solvers through representation alignment in pretrained diffusion and flow based models. While this work may have broader societal implications, we do not identify any that require specific discussion here.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not introduce new datasets, nor does it release a new generative model with increased risk of misuse. Instead, it proposes a guidance mechanism applied to existing pretrained models for solving inverse problems.

    Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have included all relevant citations on the main text and the appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This work focuses on a methodological contribution applied to existing pretrained models and standard datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: Our paper does not include research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects or user studies, and therefore does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development did not include LLMs

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.