REPRESENTATION ALIGNMENT FOR INVERSE PROBLEMS WITH DIFFUSION AND FLOW-BASED MODELS

Anonymous authors
Paper under double-blind review

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 empirically show that aligning model representations of approximate target features can substantially enhance reconstruction quality and perceptual realism. We integrate REPA into multiple state-of-the-art inverse problem solvers, and provide extensive experiments confirming that our method consistently improves reconstruction quality and realism.

1 INTRODUCTION

Pretrained diffusion and flow-based models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Lipman et al., 2023) have been at the heart of methods addressing inverse problems. These approaches perform diffusion sampling while incorporating measurement information to reconstruct the clean image (Patel et al., 2024; Chung et al., 2023; Thaker et al., 2025). Diffusion and flow-based models have pushed the boundaries in addressing inverse problems (Daras et al., 2024). However, they still struggle in cases of severe degradation and complex natural scenes with rich textures. These limitations are amplified in latent diffusion models (Rombach et al., 2022), where encoder–decoder nonlinearities introduce additional challenges. This motivates the incorporation of stronger inductive biases during inference (Rout et al., 2023; Raphaeli et al., 2025).

Recent studies have shown that diffusion models learn semantic features in their hidden states, (Li et al., 2023). The more expressive these features are the better the diffusion model performs on the generative task, (Xiang et al., 2023). Building on this insight, the seminal work of Yu et al. (2024) introduced a regularizer that aligns the internal representations of the diffusion model with those of a pretrained visual encoder, (Oquab et al., 2023). 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 motivated several follow-up works (Wang et al., 2025b; Tian et al., 2025; Yao et al., 2025; Leng et al., 2025; Wang et al., 2025a), primarily focused on improving training dynamics and convergence. However, little attention has been given to inference-time alignment for 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.

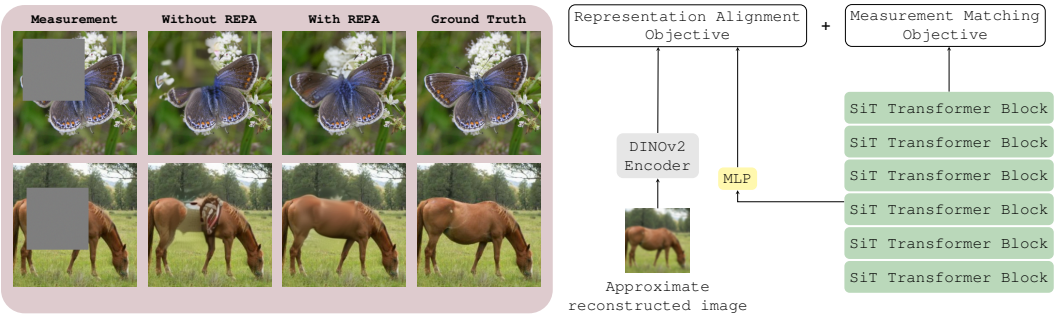


Figure 1: Overview of our proposed framework. Left: Box inpainting results, where adding REPA improves perceptual quality. Right: Alignment between diffusion features and DINOv2 embeddings.

- We integrate representational alignment into existing state-of-the-art inverse problem solvers and validate its effectiveness through extensive experiments on super-resolution, Gaussian deblurring, motion deblurring, and box inpainting, demonstrating consistent improvements over prior methods across tasks and evaluation metrics.

2 Related Work

2.1 Diffusion & Flow-based Models

Denoising-based generative models learn a data distribution by defining a forward corruption process that interpolates between data and a reference distribution (typically Gaussian) and a learned reverse process that reconstructs samples from noise (Song et al., 2021; Liu et al., 2023). In the stochastic interpolant formulation (Albergo et al., 2023), the time marginals can be expressed as

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

where α_t and σ_t correspond to decreasing and increasing function of t, respectively, with boundary conditions $\alpha_0 = \sigma_1 = 1$ and $\alpha_1 = \sigma_0 = 0$. This process admits a probability flow ODE $\dot{x}_t = v(x_t, t)$ (Lipman et al., 2023), whose solution follows the same marginals as the process in eq. 1. To enable sampling, one has to train a neural network $v_\theta(x_t, t)$ to approximate $v(x_t, t)$.

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

$$\mathrm{d}x_t = v(x_t, t)\,\mathrm{d}t - g^2(t)\, s(x_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{w}_t, \tag{2}$$

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. Diffusion models correspond to stochastic discretizations of this dynamics, while flow-based models follow the deterministic probability flow ODE (Lai et al., 2025).

2.2 Diffusion Models with Representation Guidance

Diffusion models learn rich, discriminative features in their hidden states, which are crucial for their generative performance (Xiang et al., 2023). Nevertheless, these representations fall behind those of state-of-the-art self-supervised visual encoders on downstream tasks (Siméoni et al., 2025). This gap has been identified as a bottleneck for improving generative performance (Yu et al., 2024). To address this limitation, recent works introduce representation alignment, aligning the model's internal representations with those of large pretrained encoders (Tian et al., 2025). Within the REPA framework, alignment is applied to intermediate representations of the diffusion model. Let the model consist of L transformer blocks and let $\ell \in \{1, \ldots, L\}$ denote the extraction layer. Define $h_t = \textsc{DiffEnc}(x_t, t) \in \mathbb{R}^{N \times D_2}$, where DIFFENC denotes the first ℓ transformer blocks applied to the noisy input x_t, producing N patch tokens of dimension D_2. REPA enforces patch-wise similarity between DINOv2 features $f^{[n]}_{\text{DINOv2}}(x) \in \mathbb{R}^{D_1}$ and hidden states $h_t^{[n]}$ (corresponding to the n-th patch), by maximizing

$$\textsc{Repa}(x, h_t) = \frac{1}{N} \sum_{n=1}^{N} \cos\left(f^{[n]}_{\text{DINOv2}}(x), g_\phi(h_t^{[n]}) \right), \tag{3}$$

where $g_\phi : \mathbb{R}^{D_2} \to \mathbb{R}^{D_1}$ is a learnable MLP projecting hidden states to the DINOv2 embedding space. The alignment term is optimized jointly with the standard diffusion loss.

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. (2) with the conditional score $\nabla_{x_t} \log p(x_t \mid y)$. Applying Bayes' rule, the conditional score can be decomposed as

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

The unconditional score is provided by a pretrained diffusion or flow-based model, while the likelihood term $\nabla_{x_t} \log p(y \mid x_t)$ is generally intractable, as it requires marginalizing over all possible clean signals x_0. For example, one of the most prominent approaches is *Diffusion Posterior Sampling (DPS)*, (Chung et al., 2023), which uses the following approximation:

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

3 Proposed Approach

Given a noisy observation $y \in \mathbb{R}^m$ of an unknown signal $x_0 \in \mathbb{R}^k$, our goal is to sample from $p_\theta(x_0 \mid y)$ using a pretrained diffusion- or flow-based model. We aim to apply 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_{proxy}. This representation should satisfy $c_{\text{proxy}} \equiv f_{\text{DINOv2}}(\bar{x}_0) \approx f_{\text{DINOv2}}(x_0)$. We then introduce a *tilted distribution* (Pachebat et al., 2025):

$$\widetilde{p}(x_t \mid y) \propto p(y \mid x_t)\, p_t(x_t) \exp\big(\lambda \,\text{REPA}(\bar{x}_0, h_t)\big), \tag{4}$$

where $h_t = \text{DIFFENC}(x_t, t)$ denotes the intermediate diffusion representation and $\lambda > 0$ controls the strength of the alignment term. The REPA term (Eq. equation 3) biases sampling toward representations aligned with the proxy features c_{proxy}.

On the selection of c_{proxy}. Since ground-truth images are unavailable at inference time, we construct c_{proxy} from an approximate reconstruction. We initialize the proxy using the DINOv2 features of the available observation and gradually replace it with the features of the model's current denoised estimate, namely $f_{\text{DINOv2}}(\mathbb{E}[x_0 \mid x_t])$. This approach relies on the robustness of pretrained DINOv2 features, which, as shown in our experiments, remain stable under common degradations A.3.

Extension to latent diffusion models. The same representation-alignment strategy applies in latent space. Let z_t denote the latent state at timestep t. The tilted distribution becomes

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

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. The resulting latent-space sampler is summarized in Algorithm 1, with implementation details provided in Appendix A.1 and Appendix A.2.

Algorithm 1 REPA-regularized Inverse Algorithm

Require: model u_θ, measurement y, Decoder $\mathcal{D}$ η, λ, c_{proxy}, t_{cutoff}, $z_T \sim \mathcal{N}(0, I)$

for $t = T, \ldots, 1$ **do**
  $v \leftarrow u_\theta(z_t, t)$
  $z_{t-1} \leftarrow z_t - \frac{1}{T} \cdot v + \eta \nabla_{z_t} \log p(y \mid z_t)$
  $z_{t-1} \leftarrow z_{t-1} + \lambda \nabla_{z_t} \sum_{n=1}^{N} \cos\big(c_{\text{proxy}}^{[n]}, g_\phi(\text{DiffEnc}^{[n]}(z_t, t))\big)$
  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
return z_0

Table 1: Performance comparison on ImageNet and FFHQ across inverse tasks.

Task	Method	ImageNet				FFHQ			
		LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑
4× 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	0.177	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
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
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
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

4 Experimental Results

In this section, we present experimental results demonstrating the effectiveness of our alignment regularizer. We evaluate reconstruction quality using four widely adopted metrics: PSNR (peak signal-to-noise ratio), SSIM (structural similarity index) (Wang et al., 2004), LPIPS (learned perceptual image patch similarity) (Zhang et al., 2018), and FID (Fréchet Inception Distance) (Heusel et al., 2017). Experiments are conducted on both the ImageNet and FFHQ datasets (Deng et al., 2009; Karras et al., 2019), with FFHQ images resized to 256×256. All metrics for both datasets are averaged over 100 images from their corresponding validation splits.

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-based inverse problem methods (Chung et al., 2023; Zhang et al., 2025) 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 128×128 square region. Gaussian deblurring is performed using a 61×61 kernel with standard deviation 3.0. Motion blur is simulated using a kernel with size 61×61 and intensity 0.5. For ImageNet we use additive Gaussian noise with standard deviation 0.01. For FFHQ we increase the noise level to 0.05 in order to obtain a more challenging reconstruction setting.

4.1 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 (Song et al., 2024). A detailed description of these methods and their integration with REPA is provided in Appendix A.1. For ImageNet experiments, we use the latent diffusion model trained with representation alignment from Yu et al. (2024), together with its pretrained MLP. For FFHQ, we train a representation-aligned SiT-BASE model from scratch following the same setup as Yu et al. (2024). 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.

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. These results suggest that representation alignment guides the diffusion trajectory toward semantically consistent and perceptually realistic reconstructions that better match the target image. Figure 1 presents qualitative comparisons for box inpainting, where reconstructions obtained with REPA are more faithful to the ground truth than those produced by the unregularized methods. We present more qualitative results in the Appendix A.7. In addition to improving these latent solvers, we also compare our approach with other state-of-the-art reconstruction algorithms. A detailed comparison can be found in Appendix A.6.

REFERENCES

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.

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.

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.

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*, pp. 248–255. Ieee, 2009.

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

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.

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

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*, pp. 4401–4410, 2019.

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

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.

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*, pp. 2206–2217, 2023.

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.

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.

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.

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

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.

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*, pp. 10570–10580, 2025.

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

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.

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.

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

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.

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

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*, pp. 12873–12882, 2025.

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.

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*, 2025a.

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.

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*, 2025b.

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.

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*, pp. 15703–15712, 2025.

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.

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*, pp. 20895–20905, 2025.

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*, pp. 586–595, 2018.

Table 2: Hyperparameters used for Latent DPS + REPA. ImageNet uses Gaussian noise $\sigma = 0.01$, while FFHQ uses $\sigma = 0.05$.

	ImageNet		FFHQ	
Task	κ	λ	κ	λ
Super-resolution	2.0	0.01	*0.025*	*1.25*
Gaussian deblurring	0.25	0.05	*0.05*	*0.5*
Motion deblurring	0.5	0.01	*0.05*	*0.5*
Box inpainting	0.5	0.01	*0.025*	*0.75*

Algorithm 2 Latent DPS + REPA Algorithm

Require: flow model u_θ, measurement y, pretrained encoder f, proxy representation c_{proxy}, timestep t_{cutoff}

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

A Appendix

A.1 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 κ 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 κ and the representation-alignment strength λ are tuned on a small validation set with present the best parameters found in Table 2.

A.2 Details About Resample + REPA Implementation

In Table 3, we present an adaptation of the algorithm proposed by Song et al. (2024) 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 κ and λ, 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 uses Gaussian noise $\sigma = 0.01$, while FFHQ uses $\sigma = 0.05$.

	ImageNet				FFHQ			
Task	κ	λ	**max iters**	**inner lr**	κ	λ	**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

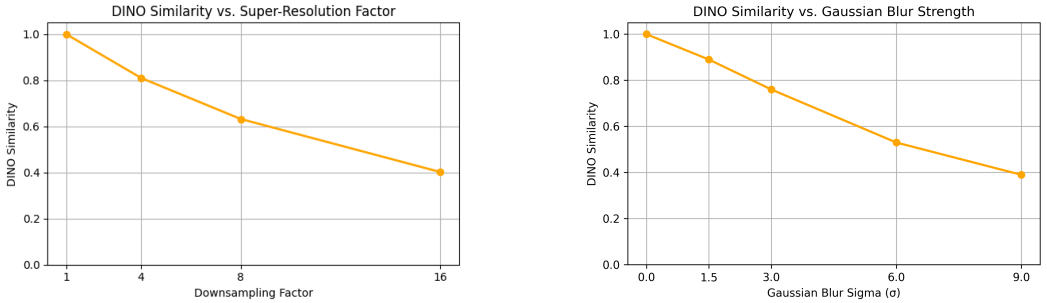
Require: flow model u_θ, measurement y, pretrained encoder f, resample steps C, parameter γ, proxy representation c_{proxy}, timestep t_{cutoff}

initialize $z_T \sim \mathcal{N}(0, I)$
for $t \in \{T, \dots, 0\}$ **do**
  $v \leftarrow u_\theta(z_t, t)$
  $\hat{z}_0 \leftarrow \mathbb{E}[z_0 \mid z_t]$
  $\Delta t \leftarrow 1/T$
  $z_{t-1} \leftarrow z_t - \Delta t \cdot v$
  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
  $z_{t-1} \leftarrow z_{t-1} - \eta \nabla_{z_t} \|y - \mathcal{A}(\mathcal{D}(\hat{z}_0))\|_2^2$
  $z_{t-1} \leftarrow z_{t-1} + \lambda \nabla_{z_t} \sum_{n=1}^{N} \cos\big(c_{\text{proxy}}^{[n]},\, g_\phi(\text{DiffEnc}^{[n]}(z_t, t))\big)$
  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.3 ROBUSTNESS OF DINOV2 REPRESENTATIONS TO CORRUPTIONS

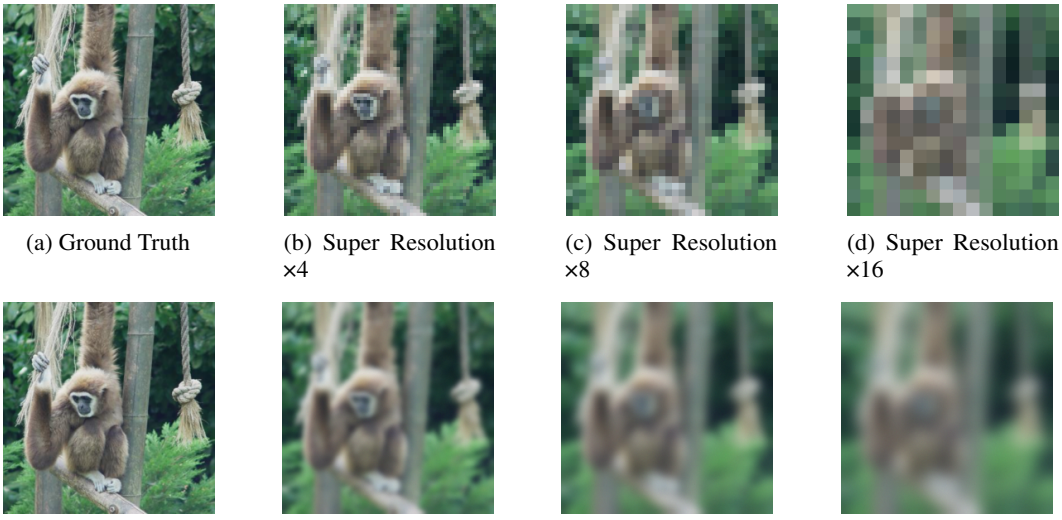
To evaluate the robustness of DINOv2 representations under various corruptions, we conduct experiments on a fixed set of 100 images. For each image, we compute the average patch similarity between its ground truth representation and that of its corrupted version. We assess this similarity across different 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 2 presents how DINOv2 representation similarity changes as the corruption severity increases for the two tasks. Figure 3 further visualizes how the pixel-space appearance of a single image changes across corruption levels. 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 2: 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 3: Visual comparison of a corrupted image used in the similarity experiments for different corruption types and severity levels.

A.4 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:

$$c_{\text{proxy}} = \text{mask} \odot f_{\text{DINOv2}}(y) + (I - \text{mask}) \odot f_{\text{DINOv2}}\big(\mathcal{D}(\mathbb{E}[z_0 \mid z_t])\big), \tag{5}$$

  where 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 yields only marginal improvements, and we therefore use the measurement-based proxy throughout.

Table 4: Performance comparison on ImageNet and FFHQ across inverse tasks. ImageNet experiments use Gaussian noise with $\sigma = 0.01$, while FFHQ uses $\sigma = 0.05$. Lower is better for LPIPS/FID, higher is better for PSNR/SSIM. Best per dataset and task in **bold**.

		ImageNet				FFHQ			
Task	**Method**	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑
4× 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	0.177	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.5 Experimental Details

We implement the inverse problem operators following the setups of Chung et al. (2023). For deblurring tasks, we follow Zhang et al. (2025) 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 Chung et al. (2023) and Zhang et al. (2025).

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.6 Comparison with State of the Art Algorithms

To further demonstrate the effectiveness of our regularizer, we compare our method against other state-of-the-art approaches. For pixel-space methods, we consider DPS (Chung et al., 2023), evaluated using the pretrained diffusion model of Dhariwal & Nichol (2021). For latent diffusion baselines, we include Latent DAPS, implemented with the conditional ImageNet latent diffusion model of Rombach et al. (2022). Table 4 summarizes the results. Our REPA regularizer consistently improves the latent solvers we evaluate and outperforms strong baselines such as DPS and Latent DAPS in most settings.

A.7 Additional Qualitative Results

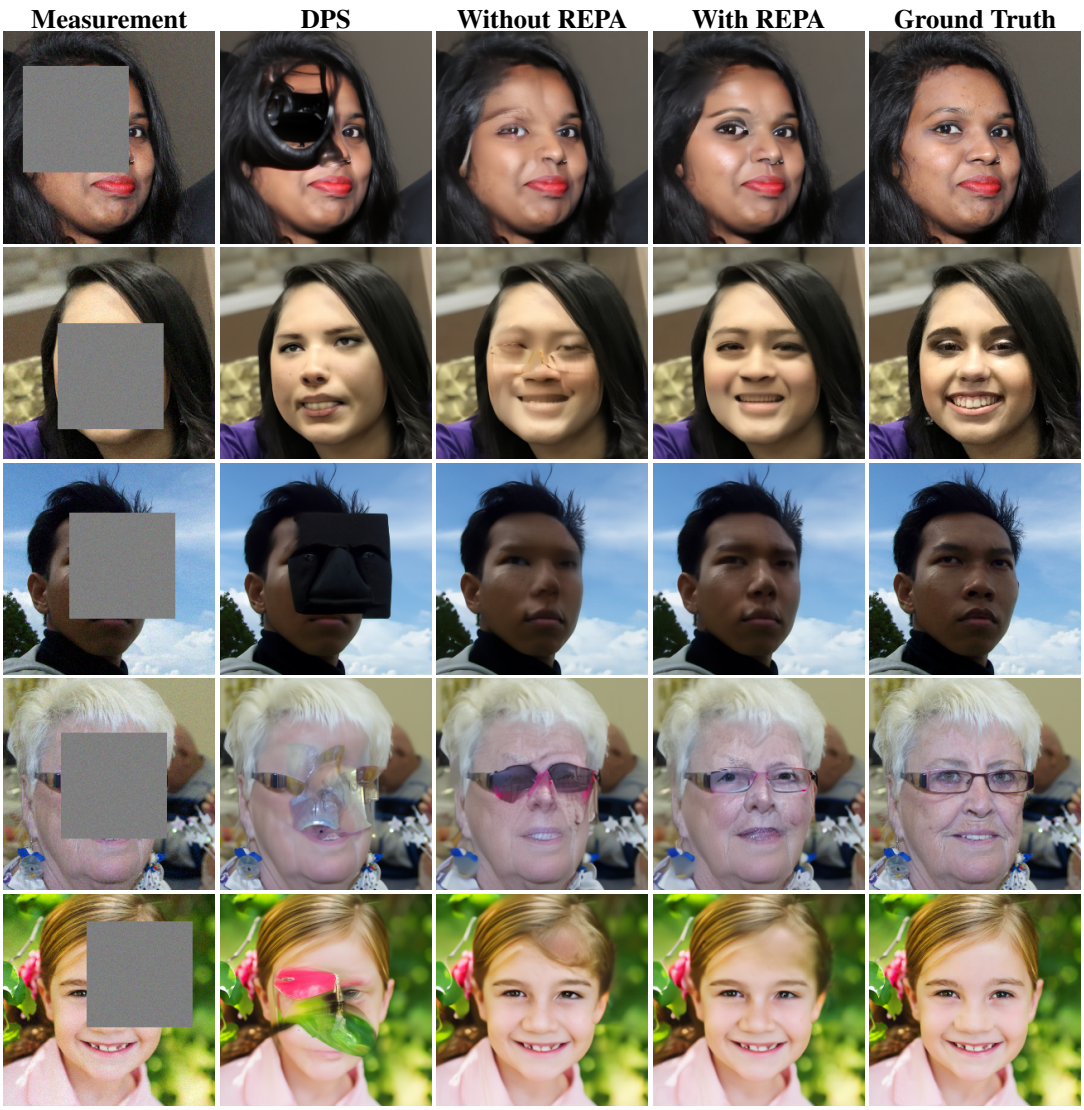


Figure 4: Qualitative comparison for box inpainting on the FFHQ dataset. Each row shows (from left to right): the measurement, the baseline method (DPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first two rows correspond to ReSample, while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

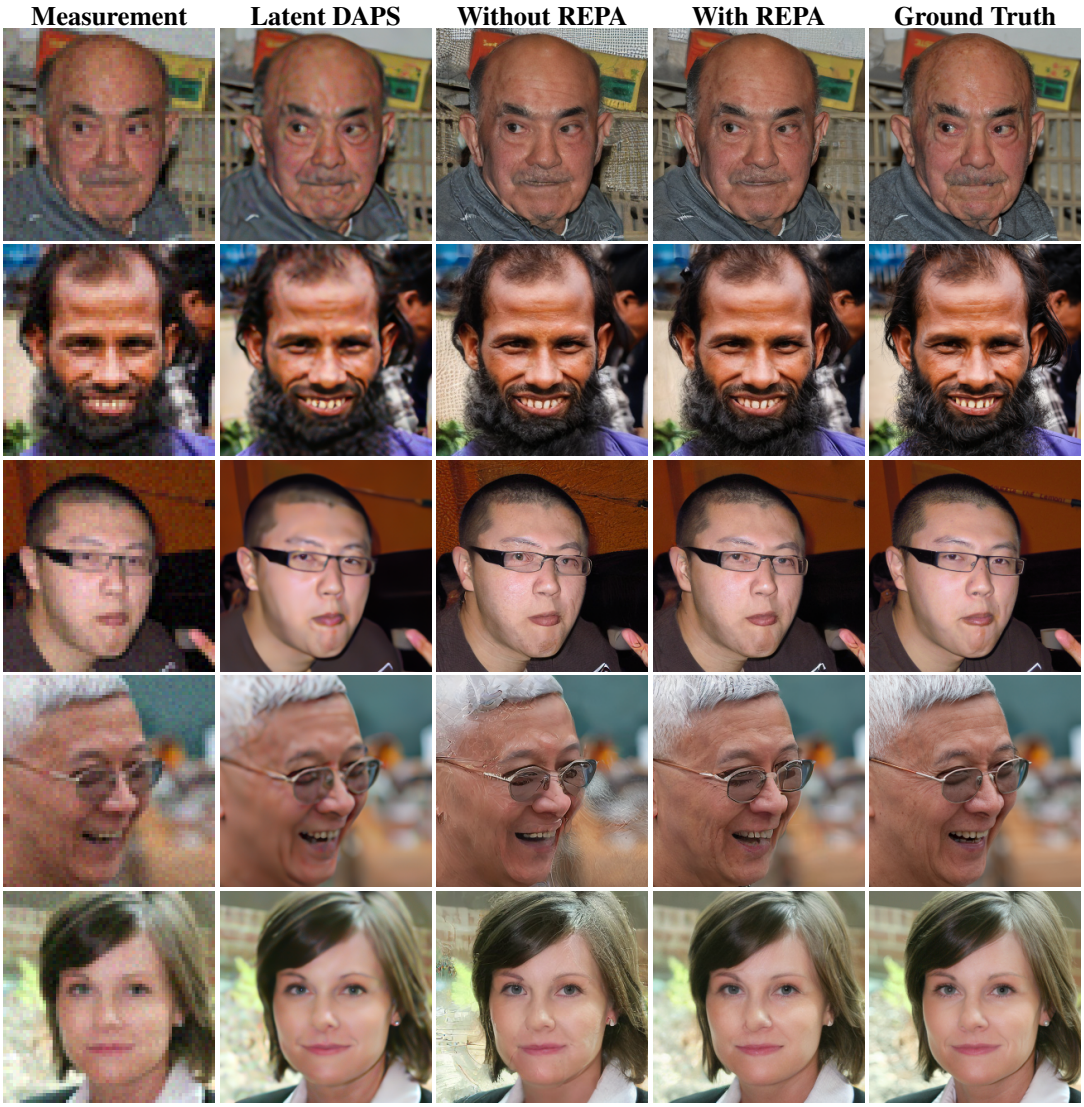


Figure 5: Qualitative comparison for $4\times$ super-resolution on the FFHQ dataset. Each row shows (from left to right): the measurement, the baseline method (Latent DAPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first two rows correspond to ReSample, while the last three correspond to Latent DPS.

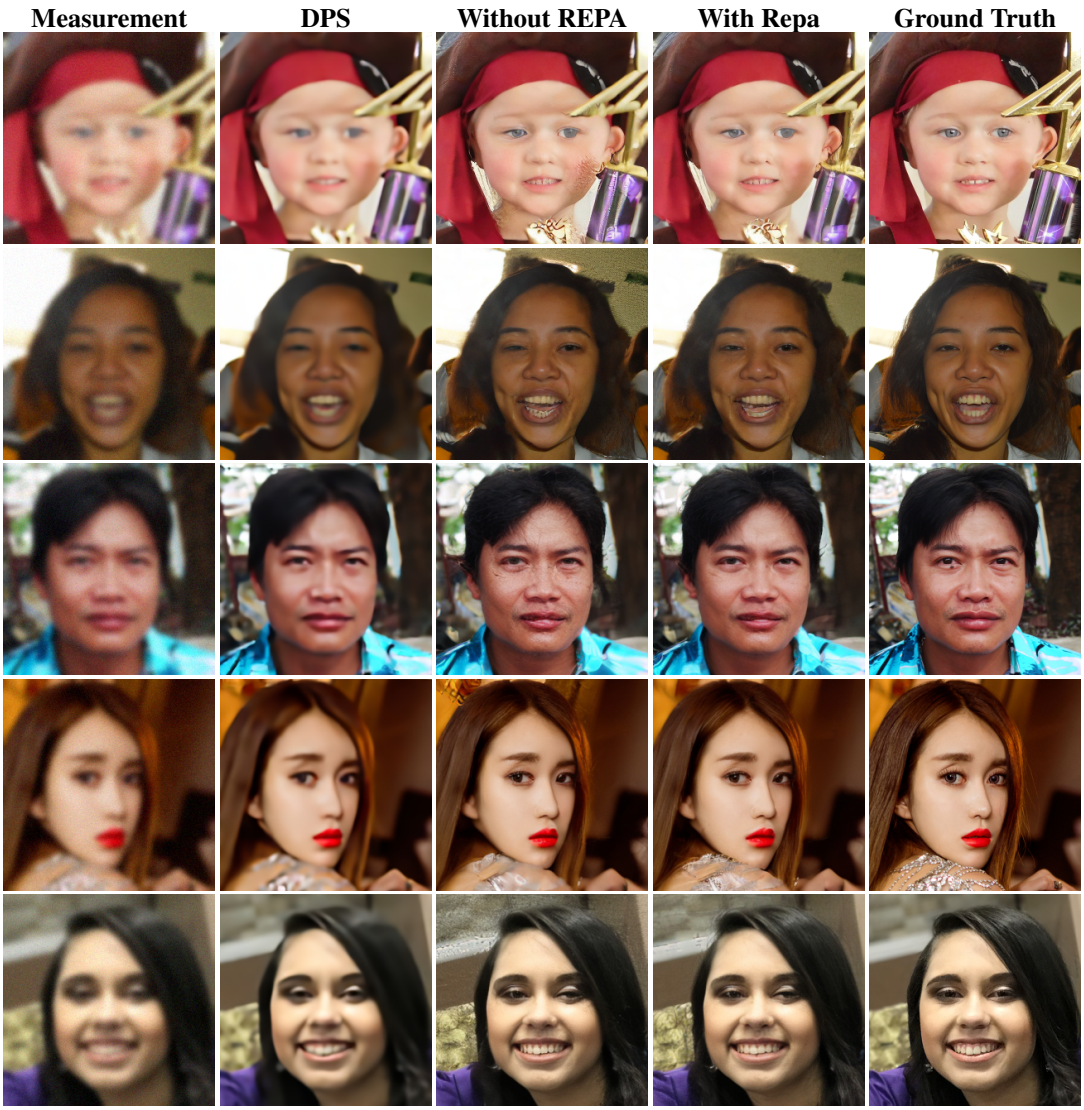


Figure 6: Qualitative comparison for Gaussian Deblurring on the FFHQ dataset. Each row shows (from left to right): the measurement, the baseline method (Latent DAPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first two rows correspond to ReSample, while the last three correspond to Latent DPS.

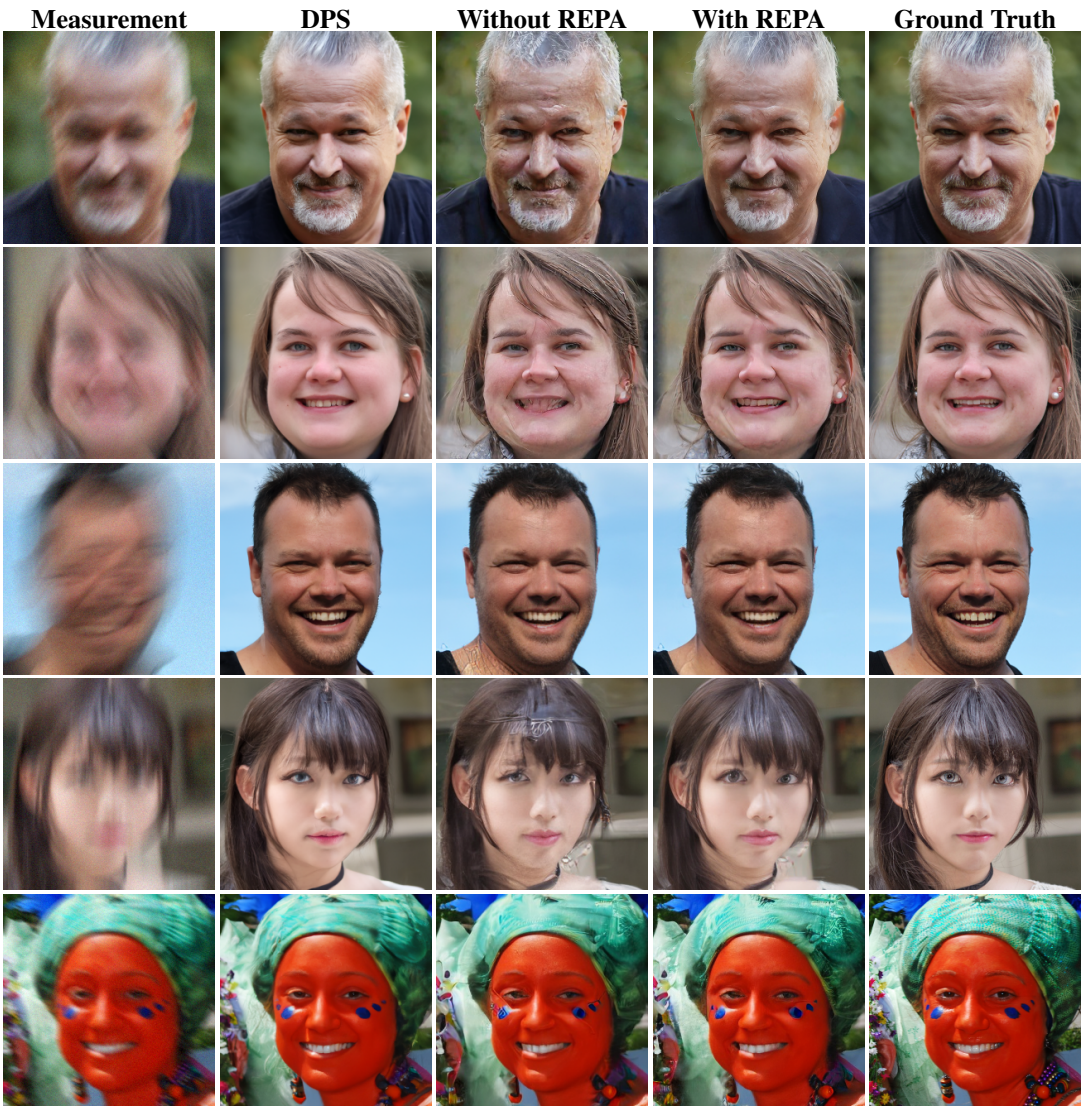


Figure 7: Qualitative comparison for Motion Deblurring on the FFHQ dataset. Each row shows (from left to right): the measurement, the baseline method (DPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first two rows correspond to ReSample, while the last three correspond to Latent DPS.

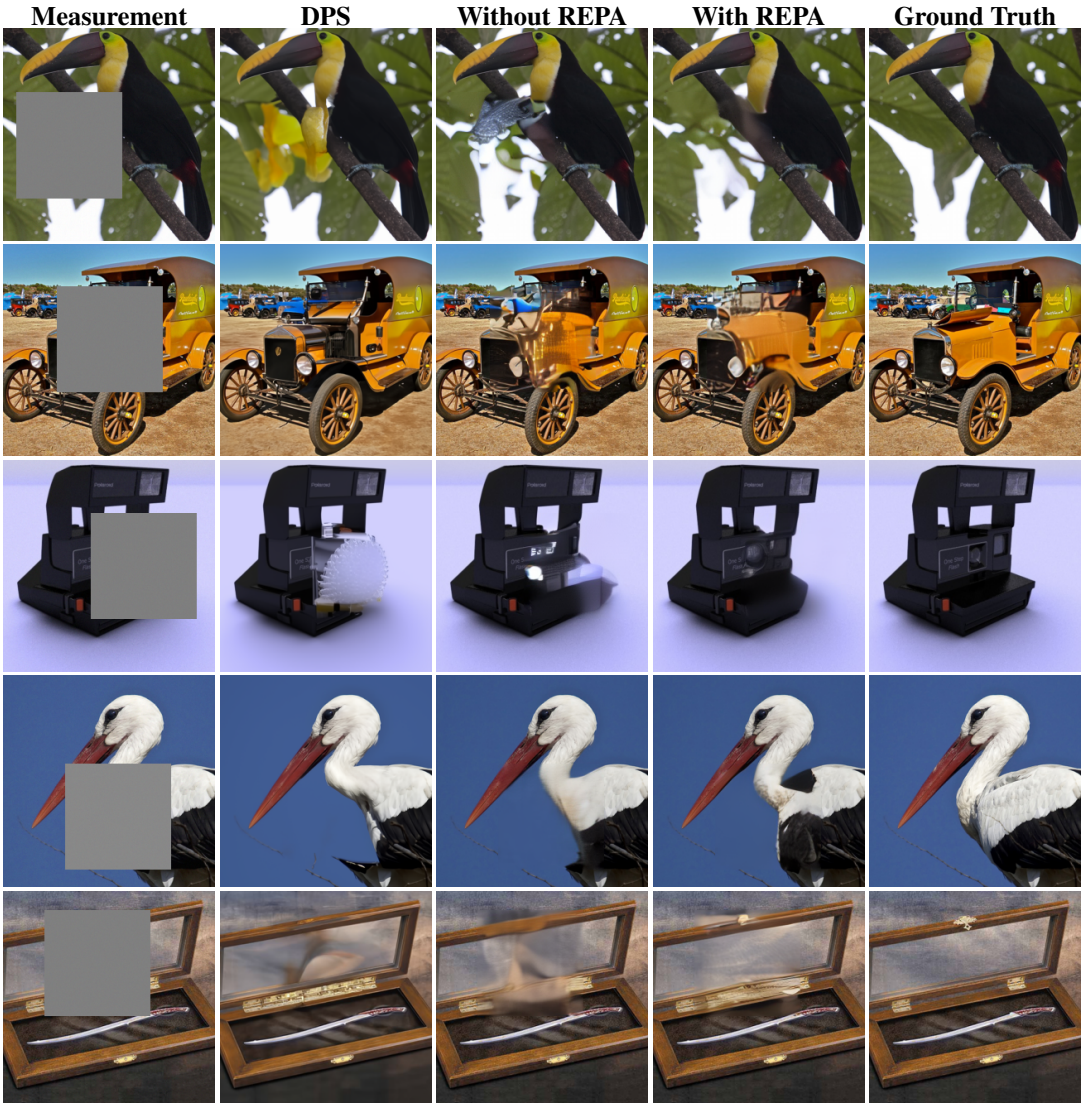


Figure 8: Qualitative comparison for box inpainting on the ImageNet dataset. Each row shows (from left to right): the measurement, the baseline method (DPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first three rows correspond to ReSample, while the last two correspond to Latent DPS.

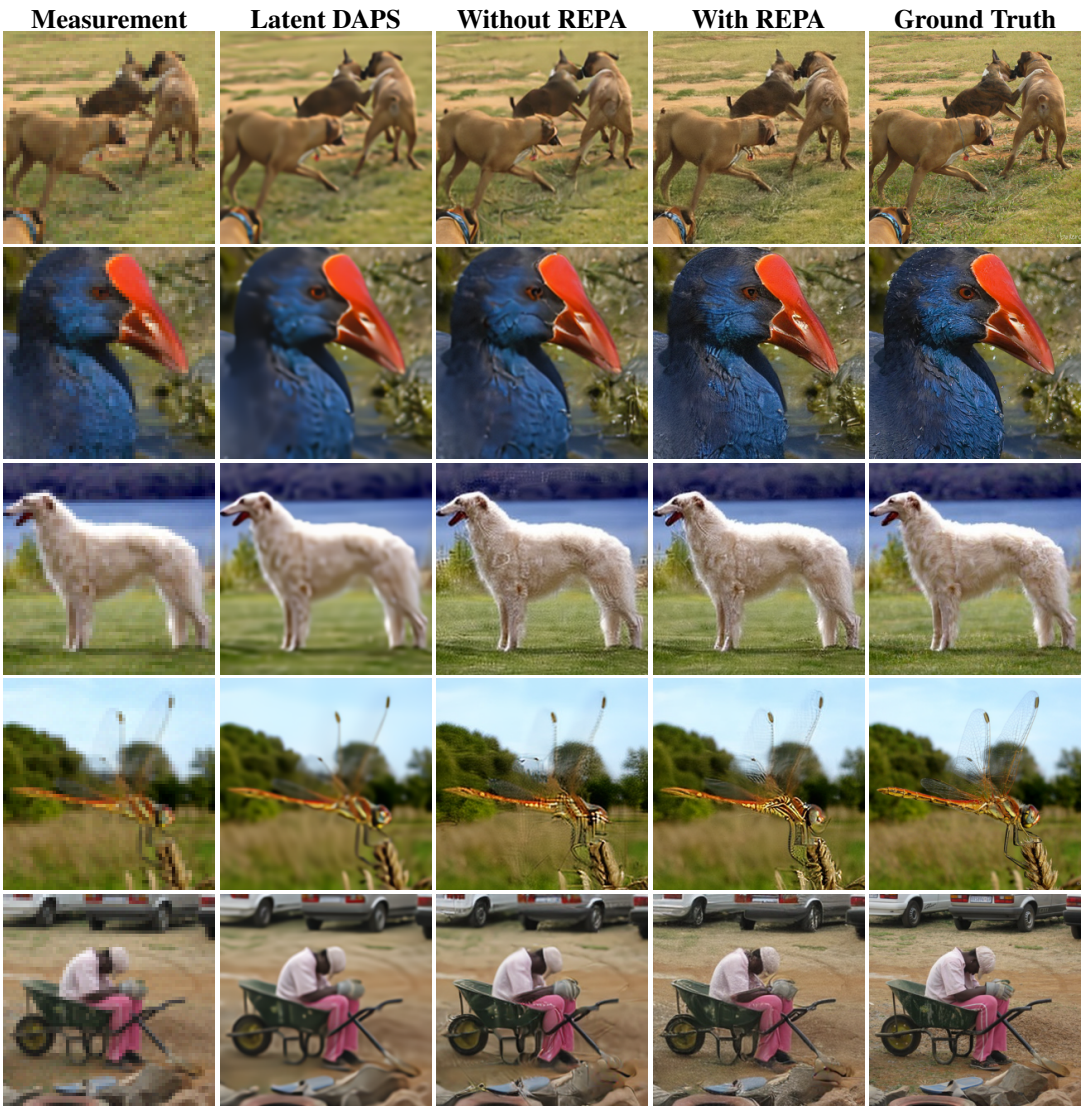


Figure 9: Qualitative comparison for Super resolution on the ImageNet dataset. Each row shows (from left to right): the measurement, the baseline method (Latent DAPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first three rows correspond to ReSample, while the last two correspond to Latent DPS.

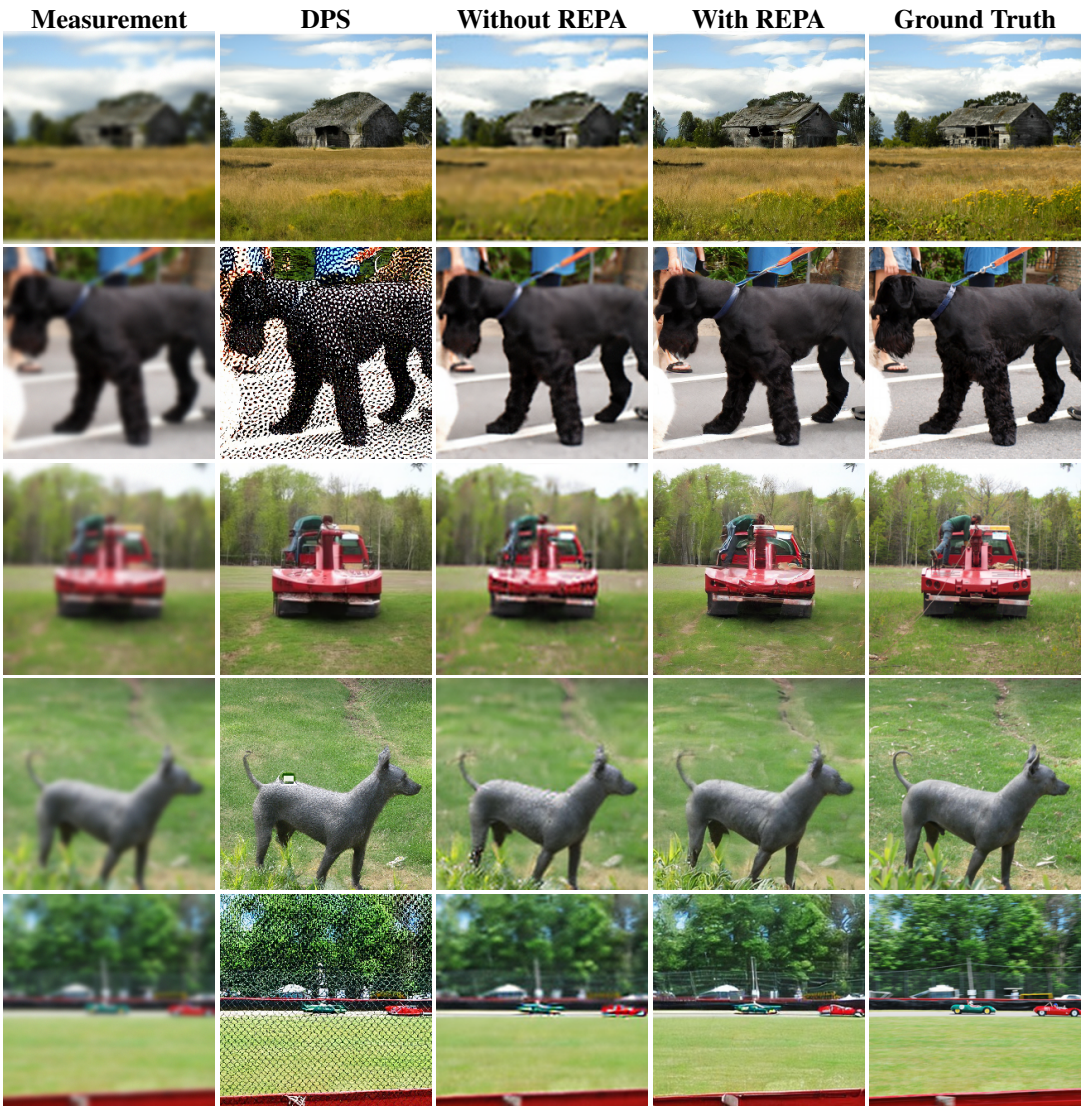


Figure 10: Qualitative comparison for Gaussian Deblurring on the ImageNet dataset. Each row shows (from left to right): the measurement, the baseline method (DPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first three rows correspond to ReSample, while the last two correspond to Latent DPS.

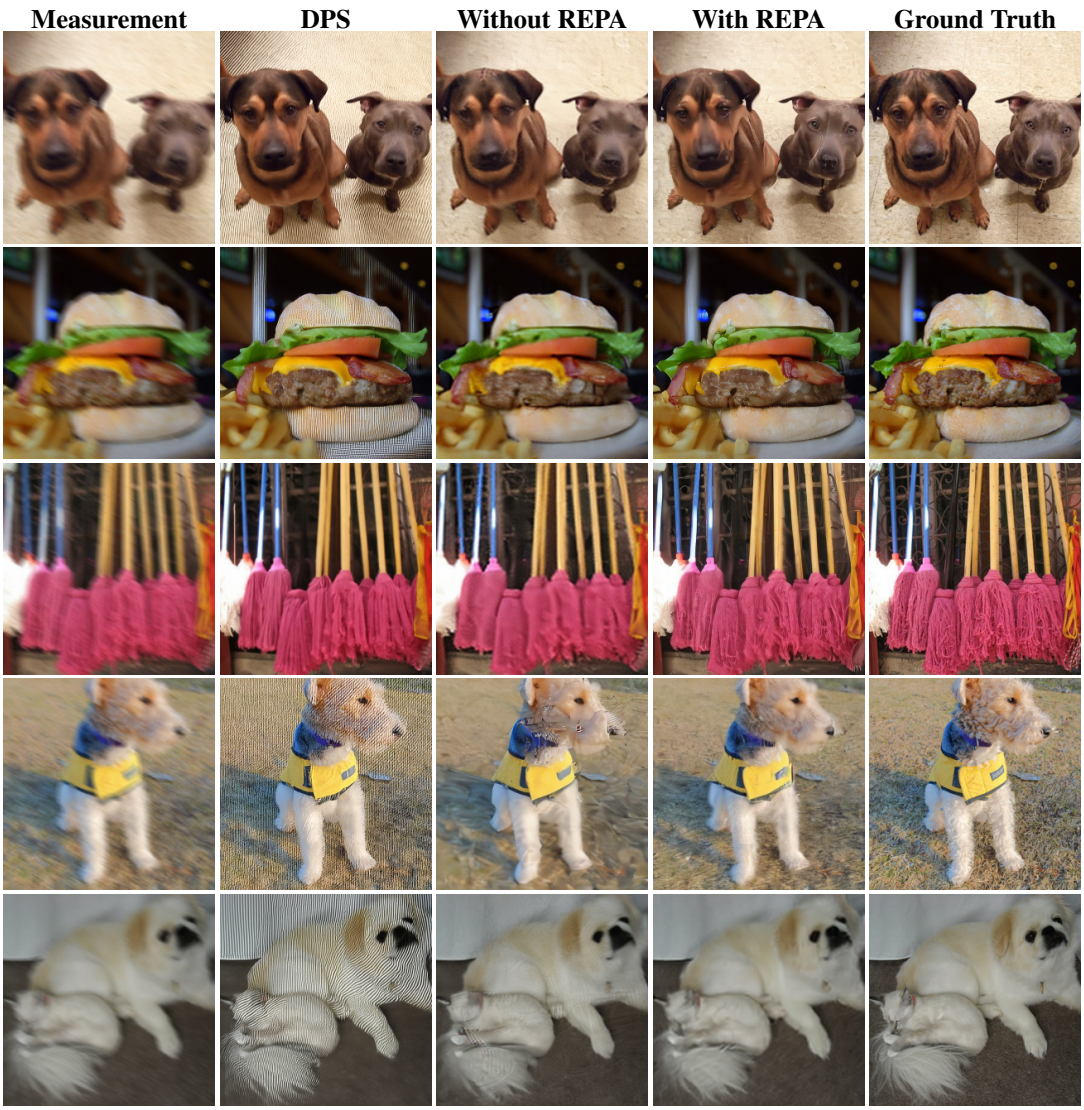


Figure 11: Qualitative comparison for Motion Deblurring on the ImageNet dataset. Each row shows (from left to right): the measurement, the baseline method (DPS), the baseline latent solver (Latent DPS or ReSample), its REPA-enhanced variant, and the ground truth. The first three rows correspond to ReSample, while the last two correspond to Latent DPS.