Solving Noisy Inverse Problems via Posterior Sampling: A Policy Gradient View-Point

Anonymous Author(s)
Affiliation
Address
email

Abstract

Solving image inverse problems (e.g., super-resolution and inpainting) requires generating a high fidelity image that matches the given input (the low-resolution image or the masked image). By using the input image as guidance, we can leverage a pretrained diffusion generation model to solve a wide range of image inversion tasks without task specific model fine-tuning. In this work, we propose diffusion policy gradient (DPG), a tractable computation method to estimate the score function given the guidance image. Our method is robust to both Gaussian and Poisson noise added to the input image, and it improves the image restoration consistency and quality on FFHQ, ImageNet and LSUN datasets on both linear and non-linear image inversion tasks (inpainting, super-resolution, motion deblur, non-linear deblur, etc.).

1 Introduction and Problem Formulation

Denoising Diffusion Probabilistic Models Ho et al. [2020], Sohl-Dickstein et al. [2015] provide tractable solutions to modeling a high quality image distribution. Their modeling and generation capabilities have been exploited in a wide range of image inverse problems Dhariwal and Nichol [2021], Blattmann et al. [2022], Rombach et al. [2021], Kawar et al. [2022], in which the goal is to generate a high quality image that matches the given input image. However, training a diffusion model from scratch is time-consuming. An alternative solution is to use the input image as guidance, and then generate the target image using a pretrained diffusion generative model through guided diffusion Ho and Salimans [2021], Dhariwal and Nichol [2021]. However, when the input guidance image is distorted by random noise and becomes inaccurate, solving image inversion problems becomes extremely challenging.

Problem Formulation We now describe the noisy image inverse problem in more details. Suppose $\mathbf{x}_0$ represents a high quality image and let $p_0(\mathbf{x}_0)$ be its distribution. Let $\mathbf{y}$ be a noisy input image, which is obtained by feeding a high quality image $\mathbf{x}_0$ through an operator $\mathcal{A}$, i.e.,

$$\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \mathbf{n}, \tag{1}$$

where $\mathbf{n}$ is the distorted random noise. The operator $\mathcal{A}$ depends on the image inversion tasks. Notice that the operator $\mathcal{A}$ is often low-rank and invertible, making the computation of the inverse of $\mathbf{y}$ impossible.

Given the noisy input $\mathbf{y}$, we can find the inverse solution $\mathbf{x}_0$ by sampling from the conditional distribution $p_0(\mathbf{x}_0|\mathbf{y}) = \frac{p_0(\mathbf{x}_0)p_0(\mathbf{y}|\mathbf{x}_0)}{p(\mathbf{y})} \propto p_0(\mathbf{x}_0)p_0(\mathbf{y}|\mathbf{x}_0)$. Sampling from $p_0(\mathbf{x}_0|\mathbf{y})$ requires information about the prior distribution $p_0(\mathbf{x}_0)$, i.e., the distribution of the high quality images. This information can be obtained via an image generative model such as diffusion models.

Solving Inverse Problem Using Pretrained Diffusion Models The forward diffusion process that turns $p_0(\mathbf{x}_0|\mathbf{y})$ into a Gaussian can be described by the following stochastic differential equation

Submitted to the DLDE-III Workshop in the 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

(SDE), i.e., $\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{w}, t \in [0, T]$, where $\beta(t) : [0, T] \mapsto \mathbb{R}^+$ is a monotonicallly increasing function and $\mathbf{w}$ is a Wiener process. Sampling from $p_0(\mathbf{x}_0|\mathbf{y})$ requires running the reverse of the forward diffusion process, i.e., running the following SDE by starting from $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$:

$$\mathrm{d}\mathbf{x} = \left[-\frac{\beta(t)}{2}\mathbf{x} - \beta(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}|\mathbf{y})\right] \mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{w}. \quad (2)$$

To generate image $\mathbf{x}_0$ for given input $\mathbf{y}$ by running equation 2, we need to compute the score function $\mathbf{s}_t(\mathbf{x}_t, \mathbf{y}) := \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})$.

Related Work There are currently two lines of work in leveraging diffusion generative models to compute the score function $\mathbf{s}_t(\mathbf{x}_t, \mathbf{y})$ and solve image inverse problems.The first line of work utilizes the low rank structure of the operator $\mathcal{A}$, and directly plugs the known information $\mathbf{y}$ into the estimation process. SDEdit, Blended Diffusion and DiffEdit Meng et al. [2022], Avrahami et al. [2022, 2023], Couairon et al. [2023] solve image inpainting and editing tasks by plugging $\mathbf{y}$ directly into the pixel space of $\mathbf{x}_0$ and then use it to predict $\mathbf{s}_t(\mathbf{x}_t, \mathbf{y})$. To solve a wider range of tasks such as super-resolution and deblur, researchers further decompose $\mathcal{A}$ using the singular value decomposition (SVD) Song et al. [2021a], Wang et al. [2023], Kawar et al. [2022], and plug the known information $\mathbf{y}$ into the spectral space of $\mathbf{x}_0$. However, those plug-in approaches can only work for linear inverse problems, and each task requires an SVD decomposition of the operator $\mathcal{A}$. To solve a wider range of non-linear image inversion problems, another line of research generate $\mathbf{x}_0$ by using the input image $\mathbf{y}$ as guidance Chung et al. [2022, 2023], Meng and Kabashima [2022], Song et al. [2023c,a], Rout et al. [2023], Song et al. [2023b], Hu et al. [2023]. Notice that the score function $\mathbf{s}_t(\mathbf{x}_t, \mathbf{y})$ can be decomposed as follows:

$$\mathbf{s}_t(\mathbf{x}_t, \mathbf{y}) := \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t, \mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t). \quad (3)$$

The first term in equation 3 is known by the pretrained diffusion model $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$. The challenge to estimate the second term, i.e., the guidance score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \mathbb{E}_{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)} [p_0(\mathbf{y}|\mathbf{x}_0)]$ in each diffusion generation step t, where $\mathbf{x}_t$ is the intermediate steps of the generation process.

Contributions. We propose a new method to estimate the score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{y}|\mathbf{x}_t)$. Our estimation can improve the image restoration quality without task specific model fine-tuning. Our contributions are summarized as follows:

(1) By viewing each noisy image $\mathbf{x}_t$ as a policy and let the predicted image $\mathbf{x}_0$ be a state that is selected by the policy, we propose diffusion policy gradient (DPG), a new method to estimate the score function given the input image $\mathbf{y}$.

(2) DPG does not need to compute a closed form psuedo-inverse or the spectral decomposition. With a pretrained diffusion generative model, we can solve a wide range of image inverse problems without model fine-tuning.

(3) Theoretically, the score function estimated by DPG is more accurate than DPS in the initial stages of the generation process. In experiments, DPG can restore more high-frequency details of the images. Quantitative evaluations on FFHQ, ImageNet and LSUN image restoration tasks show that the proposed method achieves performance improvement in both image restoration quality and consistency.

2 Methodology

2.1 Computing $\nabla_{\mathbf{x}} \log p_t(\mathbf{y}|\mathbf{x}_t)$ as Policy Gradient

We first decompose the second term $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t|\mathbf{y})$ in equation 3 as follows:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \propto \nabla_{\mathbf{x}_t} \int \underbrace{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)}_{\text{State Density Function}} \underbrace{p_0(\mathbf{y}|\mathbf{x}_0)}_{\text{Cost}} \mathrm{d}\mathbf{x}_0 =: \tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y}). \quad (4)$$

We notice that the generated image $\mathbf{x}_0$ is determined by the intermediate noisy image $\mathbf{x}_t$, and the conditional probability $p_{0|t}(\mathbf{y}|\mathbf{x}_0)$ is highly related to the reconstruction loss between $\mathbf{y}$ and the predicted image $\mathbf{x}_0$. Moreover, the score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})$ is the gradient direction of the expected loss function $\int p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)p_0(\mathbf{y}|\mathbf{x}_0)\mathrm{d}\mathbf{x}_0$. Therefore, the computation of the score function equation 4 is closely related to policy gradient in reinforcement learning, where $p_t(\mathbf{x}_t|\mathbf{x}_0)$ is the state occupation measure by choosing policy $\mathbf{x}_t$, and $p_0(\mathbf{y}|\mathbf{x}_0)$ is the cost. The following theorem enables us to compute the score function equation 4 from the policy gradient perspective, proof is provided in Appendix 5:

Theorem 1 (Leibniz Rule) *For almost all $t \in [0, T]$, we can compute the score function $\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y})$ from equation 4 as follows:*

$$\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y}) = \mathbb{E}_{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)} \left[p_0(\mathbf{y}|\mathbf{x}_0) \nabla_{\mathbf{x}_t} \log p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \right] \tag{5}$$

2.2 Implementation Details

Tractable Monte Carlo sampling Computing the score function in equation 5 requires sampling from $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)$, the closed form of which is unknown. To approximate $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)$, similar to Chung et al. [2023], Song et al. [2023c], we select a Gaussian distribution $q_{0|t}(\mathbf{x}_0|\mathbf{x}_t) = \mathcal{N}(\hat{\mathbf{x}}_0(\mathbf{x}_t), r_t^2 \mathbf{I})$ to approximate $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)$, where the mean $\hat{\mathbf{x}}_0(\mathbf{x}_t)$ is obtained by the Tweedie's estimation Efron [2011], Kim and Ye [2021], $\hat{\mathbf{x}}_0(\mathbf{x}_t) = \frac{1}{\sqrt{\overline{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{1 - \overline{\alpha}_t} \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \right)$. We select the variance $r_t = \frac{1}{C \times H \times W} \ell_{\mathbf{y}}(\mathbf{x})$, where C, H, W are the channels, height and weight of the $\mathbf{x}_0$ and $\ell_{\mathbf{y}}(\mathbf{x})$ is the reconstruction loss between $\mathbf{y}$ and the reconstructed image $\hat{\mathbf{x}}_0$. Then, by drawing N samples $\{\mathbf{x}_0^{(1)}, \cdots, \mathbf{x}_0^{(N)}\}$ from distribution $q_{0|t}(\mathbf{x}_0|\mathbf{x}_t)$, we can approximate the score function $\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y})$ in equation 5 via the Monte Carlo (MC) method $\mathbb{E}_{q_{0|t}(\mathbf{x}_0|\mathbf{x}_t)} \left[p_0(\mathbf{y}|\mathbf{x}_0) \nabla_{\mathbf{x}_t} \log q_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \right] \approx -\frac{1}{2 r_t^2 N} \sum_{i=1}^{N} \left(p_0(\mathbf{y}|\mathbf{x}_0^{(i)}) \cdot \nabla_{\mathbf{x}_t} \|\mathbf{x}_0^{(i)} - \hat{\mathbf{x}}_0(\mathbf{x}_t)\|_2^2 \right)$

Reward Shaping Similar to policy gradient in reinforcement learning, direct MC estimation of the policy gradient from suffers from high variance and low convergence rate. We leverage reward shaping Ng et al. [1999] by computing a bias term $b := \mathbb{E}_{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)} \left[p_0(\mathbf{y}|\mathbf{x}_0) \right]$ for each sample i using the leave-one-out cross-validation, $b^{(i)} := \frac{1}{N-1} \sum_{j=1, j \neq i}^{N} p_0 \left(\mathbf{y}|\mathbf{x}_0^{(i)} \right)$. We can then improve the MC estimation by $\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y}) = -\frac{1}{2 r_t^2 N} \sum_{i=1}^{N} \left((p_0(\mathbf{y}|\mathbf{x}_0^{(i)}) - b^{(i)}) \times \nabla_{\mathbf{x}_t} \|\mathbf{x}_0^{(i)} - \hat{\mathbf{x}}_0(\mathbf{x}_t)\|_2^2 \right)$

Score Function Normalization Notice that the score function computed after reward shaping contains only direction information. The exact norm of the gradient $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$ is unknown. We observe from the classifier free conditional generation experiments that the norm of the conditional score function is almost the same as the score of the unconditional generation score, and the norm is stable during the whole diffusion inference process. Therefore, we simply rescale the computed gradient into a vector with norm C, i.e., assume that $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx C \cdot \frac{1}{\|\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y})\|_2^2} \tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y})$ and plug it into equation 3 to compute the score function $\mathbf{s}_t(\mathbf{x}_t, \mathbf{y})$, i.e.,

$$\mathbf{s}_t(\mathbf{x}_t, \mathbf{y}) \approx \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + C \cdot \frac{\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y})}{\|\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y})\|_2^2}. \tag{6}$$

Using equation 6, we can solve the image inversion problems with the standard DDPM sampling method, which is displayed in Algorithm 1

Algorithm 1 Diffusion Policy Gradient (DPG)

Require: $T, \mathbf{y}, \ell(\mathbf{y}, \mathcal{A}(\cdot))$
  $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
  for $t = T - 1$ to 0 **do**
    $\hat{\mathbf{x}}_0 \leftarrow \frac{1}{\sqrt{\overline{\alpha}_t}} (\mathbf{x}_t + (1 - \overline{\alpha}_t \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t))$
    $r_t \leftarrow \frac{1}{C \times H \times W} \ell_{\mathbf{y}}(\mathbf{x})$.
    $\boldsymbol{\xi}^{(i)} \sim \mathcal{N}(0, \mathbf{I}), \mathbf{x}_0^{(i)} \leftarrow \hat{\mathbf{x}}_0 + r_t \boldsymbol{\xi}^{(i)}, i = 1, \cdots, N_{\mathrm{mc}}$.
    $b^{(i)} \leftarrow \frac{1}{N_{\mathrm{mc}} - 1} \sum_{j=1, j \neq i}^{N_{\mathrm{mc}}} p_0(\mathbf{y}|\mathbf{x}_0^{(i)})$
    $\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y}) \leftarrow \frac{1}{N_{\mathrm{mc}}} \sum_{i=1}^{N_{\mathrm{mc}}} (p_0(\mathbf{y}|\mathbf{x}_0^{(i)}) - b^{(i)}) \nabla_{\mathbf{x}_t} \left(-\|\mathbf{x}_0^{(i)} - \hat{\mathbf{x}}_0(\mathbf{x}_t)\|_2^2 \right)$
    $\mathbf{s}_t(\mathbf{x}_t, \mathbf{y}) \leftarrow \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) + C \frac{\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y})}{\|\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y})\|_2^2}$
    $\mathbf{x}_{t-1} \leftarrow$DDPM$(\mathbf{x}_t, \mathbf{s}_t(\mathbf{x}_t, \mathbf{y}))$.
  end for
  Return image $\mathbf{x}_0$

3 Experiments

Experiment Setup We test the performance of our proposed algorithm on three datasets: the FFHQ 256×256 dataset Karras et al. [2019], the ImageNet dataset Deng et al. [2009] and the LSUN-Bedroom dataset Yu et al. [2015]. We consider four types of image inverse tasks: (1) Inpainting with a 128×128 masks placed randomly on the figure; (2) 4×super-resolution with average pooling; (3) Gaussian deblur with kernel size 61 × 61 and standard deviation of 3.0; (4) Motion deblur with kernel size of 61 and intensity value 0.5 generated by[1].

[1]`https://github.com/LeviBorodenko/motionblur`

We consider that the input image is noisy, i.e., Gaussian noise with variance $\sigma_{\mathbf{y}} = 0.05$ or Poisson noise with rate $\lambda = 1.0$ is added on the input image. For FFHQ experiments, we use the pretrained model from Chung et al. [2023] (trained on 4.9k images on FFHQ) and test the performance of 1k validation set; For Imagenet and LSUN experiments, we use the unconditional Imagenet and LSUN-Bedroom 256×256 generation model from Dhariwal and Nichol [2021]. We evaluate the performance on 1k ImageNet validation set images [2] and the full LSUN-Bedroom validation set.

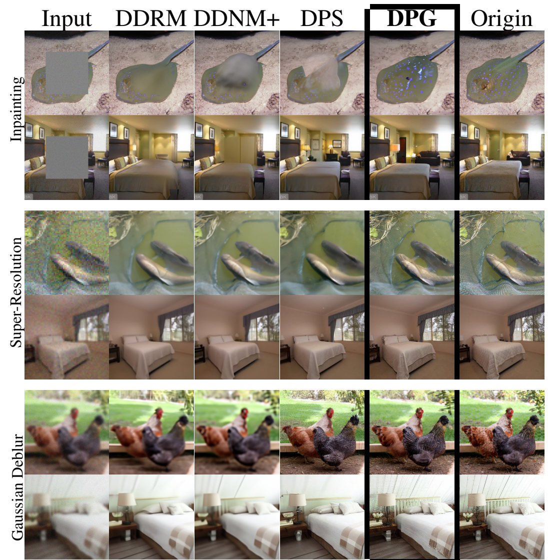


Figure 1: Results on solving linear noisy inverse problems with Gaussian noise $\sigma_{\mathbf{y}} = 0.05$ on ImageNet and LSUN Dataset.

Evaluations We measure both the image restoration quality and consistency compared with the ground-truth image. For image restoration quality, we compute the Fréchet inception distance (FID) between the restored images and the ground truth images; For image restoration consistency, we compute the LPIPS score Zhang et al. [2018] (VGG Net) between the restored image and the ground truth image. Quantitative evaluation results are displayed in Table 1. Selected image restoration samples when the observation noise are Gaussian and Poisson are displayed in Fig. 1 and Fig. 5 in the Appendix 8.2. We compare the performance with the following methods: Denoising Diffusion Null Space models (DDNM+) Wang et al. [2023], Diffusion Posterior Sampling (DPS) Chung et al. [2022] and the Denoising Diffusion Restoration Models (DDRM) Kawar et al. [2022]. The key paremeters for different methods are displayed in Appendix 7.

Analysis First, the FID and LPIPS score of our proposed DPG method is smaller than the DPS method in most tasks, indicating that DPG has a better image restoration quality than DPS method. This is because the estimation of the score function by DPG is more accurate than DPS, especially in the initial stages of the diffusion generation process. Therefore, the shape and structure of the image can be recovered in an earlilier stage of the diffusion process, this gives room to recover high frequency details in later stage of the image generation. In Appendix 6 we will analyze this observation both theoretically and empirically. Notice that DDNM+ and DDRM uses a plug-in estimation, i.e., the known pixels in $\mathbf{y}$ are directly used in the generation process. Therefore, the input noise added on the input degrade the image restoration quality. From Fig. 1, DPG recovers more high frequency details of the ground truth image, and therefore receives a smaller LPIPS and FID score in most image inverse tasks compared with DDNM+ and DDRM in most experiments. More results on Poisson input noise and non-linear image inverse tasks can be found Appendix 8.

4 Conclusions

In this paper, we proposed a new method to estimate the score function for solving image inverse problems. Our method is robust when the input image is perturbed by random noise, and can be used for solving non-linear inverse problems such as non-linear deblur. Experiments demonstrate that the proposed method can improve image restoration quality in both human eye evaluation and quantitative metrics. In the future, we will test the performance of DPG method on non-differentiable image inverse tasks such as JPEG restoration.

Table 1: Quantitative Results on Linear Inverse Problems with Gaussian Noise (**Bold**: best; underlined: second best)

Method	Inpainting		Super-Resolution		Deblur (Gauss)		Deblur (Motion)	
	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓
FFHQ 1k Validation Set								
DPG	**22.44**	0.181	**22.49**	**0.214**	**22.29**	**0.216**	**24.44**	**0.223**
DPS	33.12	0.168	39.35	**0.214**	44.05	0.257	39.02	0.242
DDRM	27.47	**0.172**	62.15	0.294	74.92	0.332	N/A	N/A
DDNM+	27.34	0.173	46.13	0.260	63.19	0.301	N/A	N/A
ImageNet 1k Validation Set								
DPG	**41.09**	0.266	**31.02**	**0.293**	**34.43**	**0.314**	**36.15**	**0.343**
DPS	45.95	0.267	43.60	0.340	62.65	0.434	56.08	0.386
DDRM	50.94	0.246	51.77	0.355	72.49	0.345	N/A	N/A
DDNM+	50.50	0.246	51.08	0.362	71.74	0.410	N/A	N/A
LSUN-Bedroom Validation Set								
DPG	**34.32**	0.218	**31.44**	**0.262**	**38.72**	**0.277**	**34.44**	**0.284**
DPS	35.91	0.218	37.42	0.284	48.10	0.320	50.09	0.358
DDRM	37.61	0.205	50.96	0.310	59.04	0.353	N/A	N/A
DDNM+	37.03	**0.204**	50.15	0.296	74.40	0.336	N/A	N/A

[2] https://github.com/XingangPan/deep-generative-prior/blob/master/scripts/imagenet_val_1k.txt

References

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

Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Trans. Graph.*, 42(4), jul 2023. ISSN 0730-0301. doi: 10.1145/3592450. URL `https://doi.org/10.1145/3592450`.

Andreas Blattmann, Robin Rombach, Kaan Oktay, and Björn Ommer. Retrieval-augmented diffusion models, 2022. URL `https://arxiv.org/abs/2204.11824`.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=nJJjv0JDJju`.

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. URL `https://openreview.net/forum?id=OnD9zGAGT0k`.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=3lge0p5o-M-`.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

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

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL `https://openreview.net/forum?id=qw8AKxfYbI`.

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

Yujie Hu, Yinhuai Wang, and Jian Zhang. Dear-gan: Degradation-aware face restoration with gan prior. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9):4603–4615, 2023. doi: 10.1109/TCSVT.2023.3244786.

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.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022.

Kwanyoung Kim and Jong Chul Ye. Noise2score: Tweedie's approach to self-supervised image denoising without clean images. In *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=ZqEUs3sTRU0`.

Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. *arXiv preprint arXiv:2305.04391*, 2023.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=aBsCjcPu_tE`.

Xiangming Meng and Yoshiyuki Kabashima. Diffusion model based posterior sampling for noisy linear inverse problems. *arXiv preprint arXiv:2211.12343*, 2022.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 278–287, San Francisco, CA, USA, 1999.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alexandros G Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. *arXiv preprint arXiv:2307.00619*, 2023.

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.

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, 2023a.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023b. URL `https://openreview.net/forum?id=9_gsMA8MRKQ`.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32483–32498. PMLR, 23–29 Jul 2023c. URL `https://proceedings.mlr.press/v202/song23k.html`.

Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *NeurIPS 2021 Workshop on Deep Learning and Inverse Problems*, 2021a. URL `https://openreview.net/forum?id=4rFAhgrA0lA`.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL `https://openreview.net/forum?id=PxTIG12RRHS`.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *The Eleventh International Conference on Learning Representations*, 2023.

Allan G Weber. The usc-sipi image database: Version 5. *http://sipi. usc. edu/database/*, 2006.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

5 Proof of Theorem 1

To prove Theorem 1, we need to verify the following three condition holds for density function $p_t(\mathbf{x}_t|\mathbf{y})$: For fixed $\mathbf{x}_0$ and any time $\tau < T$, if the conditional density function $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t = \mathbf{x})$ satisfy the following conditions: (1) function $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t = \mathbf{x})$ is a Lebesgue-integrable of t for each $\mathbf{x}$; (2) the gradient $\nabla_{\mathbf{x}_t} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t), \forall \mathbf{x}$ exists for almost all $t \in [0, T]$; (3) there is an integral function $g(t)$ so that $\|\nabla_{\mathbf{x}_t} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\| \leq g(t)$ We will first verify each condition in Theorem 1 respectively and then provide detailed derivations of equation 5.

(1) To show function $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)$ is integrable of t, we show that $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)$ is bounded. Recall that $p_0(\mathbf{x}_0)$ is the probability density function of the high-quality images and let $\mu_0(\mathrm{d}\mathbf{x}_0)$ be the probability measure. Then density function $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)$ can be computed by:

$$p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) = \frac{\mu_0(\mathrm{d}\mathbf{x}_0) p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)}{\mathrm{d}\mathbf{x}_0 \int p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\mu_0(\mathrm{d}\mathbf{x}_0)}. \tag{7}$$

According to [Song et al., 2021b, Eq. (29)], distribution $p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\sqrt{\overline{\alpha}(t)}\mathbf{x}_0, (1 - \overline{\alpha}(t))\mathbf{I})$ is Gaussian, hence $p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)$ is bounded. Therefore, $p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) < 1, \forall t$ is bounded on $[0, T]$ and is hence Lebesgue-integrable of t.

(2) The gradient of conditional density function $\nabla_{\mathbf{x}_t} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)$ can be decomposed by:

$$\nabla_{\mathbf{x}_t} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) = \nabla_{\mathbf{x}_t}\left(\frac{p_{t,0}(\mathbf{x}_t, \mathbf{x}_0)}{p_t(\mathbf{x}_t)}\right) = \frac{1}{p_t(\mathbf{x}_t)^2}\left(\nabla_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{x}_0) \cdot p_t(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t) \cdot p(\mathbf{x}_t|\mathbf{x}_0)\right). \tag{8}$$

According to [Song et al., 2021b, Eq. (29)], distribution $p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\sqrt{\overline{\alpha}(t)}\mathbf{x}_0, (1 - \overline{\alpha}(t))\mathbf{I})$ is Gaussian, therefore, $p_t(\mathbf{x}_t)$ is non-zero $\forall \mathbf{x}_t \in \mathbb{R}, t < T$ and the gradient $\nabla_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{x}_0)$ exists for all $\mathbf{x}_t, \forall t$. It then remains to prove that gradient $\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t)$ exists for almost all $t \in [0, T]$. Since $p_t(\mathbf{x}_t) = \mathbb{E}_{p_0}\left[p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\right]$ and density function $p(\mathbf{x}_t|\mathbf{x}_0)$ is a Gaussian, as function $p(\mathbf{x}_t|\mathbf{x}_0)$ is continuous on $\mathbf{x}_t$, function $p_t(\mathbf{x}_t)$ is continuous for all $t \in [0, T)$ and hence $\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t)$ exists for almost all $t \in [0, T]$.

(3) According to equation 8, the norm

$$\begin{aligned}
\|\nabla_{\mathbf{x}_t} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\|_2^2 =& \|p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\nabla_{\mathbf{x}_t} \log p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\|_2^2 \\
=& \|p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\left(\nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right)\|_2^2 \\
\leq& \|p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\|_2^2 + \|p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\|_2^2.
\end{aligned} \tag{9}$$

Notice that $p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)$ is a Gaussian distribution, therefore the gradient can be computed by:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) =& \nabla_{\mathbf{x}_t} \log\left(\frac{1}{(2\pi(1 - \overline{\alpha}(t))^{d_{\mathbf{x}}/2}} \exp\left(-\frac{\|\mathbf{x}_t - \mathbf{x}_0\|_2^2}{2(1 - \overline{\alpha}(t))}\right)\right) \\
=& -\frac{1}{1 - \overline{\alpha}(t)}(\mathbf{x}_t - \mathbf{x}_0)
\end{aligned} \tag{10}$$

For notation simplicity, denote $\mathbf{z} = \frac{1}{\sqrt{1 - \overline{\alpha}(t)}}(\mathbf{x}_t - \mathbf{x}_0)$. Then the norm of gradient $\nabla_{\mathbf{x}_t} p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)$ can be upper bounded by:

$$\|p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\nabla_{\mathbf{x}_t} \log p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\|_2^2 \overset{(a)}{\leq} \|\frac{1}{1 - \overline{\alpha}(t)}(\mathbf{x}_t - \mathbf{x}_0)\|_2^2 \leq \frac{2}{1 - \overline{\alpha}(t)} d_{\mathbf{x}}, \tag{11}$$

Since $\overline{\alpha}(t) < 1$ is continuous, function $\frac{2}{1 - \overline{\alpha}(t)} d_{\mathbf{x}}$ is continuous and bounded on $[0, \tau), \forall \tau < T$.

Consider that $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log \mathbb{E}_{p_0}\left[p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\right]$. Since $p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)$ is a Gaussian, we can exchange the integral and gradient operator and then obtain $\nabla_{\mathbf{x}_t} \log \mathbb{E}_{p_0}\left[p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\right] = \frac{1}{\mathbb{E}_{p_0}\left[p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\right]}\mathbb{E}_{p_0}\left[\nabla_{\mathbf{x}_t} p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\right]$. Then according to equation 11, the gradient of $\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t)$ can also be upper bounded by:

$$\frac{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)}{\mathbb{E}_{p_0}\left[p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\right]}\|\mathbb{E}_{p_0}\left[\nabla_{\mathbf{x}_t} p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)\right]\|_2^2 \leq \frac{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)}{p(\mathbf{x}_t)} \frac{2}{1 - \overline{\alpha}(t)} d_{\mathbf{x}}. \tag{12}$$

Plugging equation 11 and equation 12 into equation 9, we can upper bound $\|\nabla_{\mathbf{x}_t} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\|_2^2$ as follows:

$$\|\nabla_{\mathbf{x}_t} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)\|_2^2 \leq 2C_1(1-\overline{\alpha}(t))^{-(d_{\mathbf{x}}+1)/2} =: g(t). \tag{13}$$

Since function $g(t)$ is continuous and bounded on $[0,\tau], \forall \tau < T$, function $g(t)$ is integrable. Therefore, we can apply the Leibniz rule and compute the score function $\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y})$ in equation **??** as follows:

$$\begin{aligned}
\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y}) =& \nabla_{\mathbf{x}_t} \left(\int p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) p_0(\mathbf{y}|\mathbf{x}_0) \mathrm{d}\mathbf{x}_0 \right) \\
\overset{(b)}{=}& \int \nabla_{\mathbf{x}_t} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) p_0(\mathbf{y}|\mathbf{x}_0) \mathrm{d}\mathbf{x}_0 \\
=& \int p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \left(\frac{1}{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)} \nabla_{\mathbf{x}_t} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \right) p_0(\mathbf{y}|\mathbf{x}_0) \mathrm{d}\mathbf{x}_0 \\
\overset{(c)}{=}& \int p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \nabla_{\mathbf{x}_t} \log p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) p_0(\mathbf{y}|\mathbf{x}_0) \mathrm{d}\mathbf{x}_0 \\
=& \mathbb{E}_{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)} \left[p_0(\mathbf{y}|\mathbf{x}_0) \nabla_{\mathbf{x}_t} \log p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) \right],
\end{aligned} \tag{14}$$

where equation (b) is obtained by exchanging the integration and gradient operator; equation (c) is obtained because $\nabla_{\mathbf{x}_t} \log p_{0|t}(\mathbf{x}_0|\mathbf{x}_t) = \frac{1}{p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)} p_{0|t}(\mathbf{x}_0|\mathbf{x}_t)$.

6 Analysis of Score Function Estimation Accuracy

First, we will present the following corollary:

Corollary 1 *When $r_t \to 0$ and $N \to \infty$, if $p_0(\mathbf{y}|\mathbf{x}_0)$ is a Gaussian distribution, then the score function 5 is approximately*

$$\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y}) = \frac{1}{2\sigma_{\mathbf{y}}^2 r_t} p_0(\mathbf{y}|\mathbf{x}_0) \nabla_{\mathbf{x}_t} \ell(\mathbf{y}, \mathcal{A}(\hat{\mathbf{x}}_0(\mathbf{x}_t))), \tag{15}$$

whose direction is the same of the score function in DPS Chung et al. [2023].

Proof of Corollary 1 is provided in Section 6.1. Corollary 1 shows that the score function $\tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y}) \approx \nabla_{\mathbf{x}_t} \ell(\mathbf{y}, \mathcal{A}(\hat{\mathbf{x}}_0(\mathbf{x}_t)))$ used by DPS Chung et al. [2023] is accurate when $r_t \to 0$, i.e., in later-stages of the diffusion generation process. However, in initial stages of the image generation (i.e., t is large), the score function obtained by DPS is inaccurate, therefore, as is shown in Fig. 3, the reconstruction loss $\ell(\mathbf{y}, \mathcal{A}(\hat{\mathbf{x}}_0(\mathbf{x}_t))$ by running the DPS algorithm in the initial image generation stages (i.e., $t \geq 750$) larger compared with our proposed DPG method. Fig. 2 plots the intermediate recoverved figures during the diffusion process. Since DPG has a more accurate estimation of the guidance score function, the shape the sketch of the image is recovered at an earlier stage compared with the DPG method (i.e., at time step $t = 900$, noisy image generated by DPG has the sketch of the chicken, while the image generated by DPS is blank.)

6.1 Proof of Corollary 1

$$\begin{aligned}
& \tilde{\mathbf{s}}_t(\mathbf{x}_t, \mathbf{y}) \\
=& \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)} \left[p_0(\mathbf{y}|\mathbf{x}_0) \nabla_{\mathbf{x}_t} \left(-\frac{1}{2r_t^2} \|\mathbf{x}_0 - \hat{\mathbf{x}}_0(\mathbf{x}_t)\|_2^2 \right) \right] \\
=& \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_t)} \left[-\frac{1}{r_t^2} p_0(\mathbf{y}|\mathbf{x}_0) \left((\hat{\mathbf{x}}_0(\mathbf{x}_t) - \mathbf{x}_0)^T \frac{\partial \hat{\mathbf{x}}_0(\mathbf{x}_t)}{\partial \mathbf{x}_t} \right)^T \right] \\
\overset{(a)}{=}& \mathbb{E}_{\boldsymbol{\xi} \sim \mathcal{N}(0, \mathbf{I})} \left[-\frac{1}{r_t^2} p_0(\mathbf{y}|\hat{\mathbf{x}}_0(\mathbf{x}_t) + r_t \boldsymbol{\xi}) \left(\boldsymbol{\xi}^T \frac{\partial \hat{\mathbf{x}}_0(\mathbf{x}_t)}{\partial \mathbf{x}_t} \right)^T \right] \\
\overset{(b)}{\approx}& \mathbb{E}_{\boldsymbol{\xi} \sim \mathcal{N}(0, \mathbf{I})} \left[-\frac{1}{r_t^2} \left(p_0(\mathbf{y}|\hat{\mathbf{x}}_0) + r_t \nabla_{\mathbf{x}_0}^T p_0(\mathbf{y}|\hat{\mathbf{x}}_0) \boldsymbol{\xi} \right) \left(\boldsymbol{\xi}^T \frac{\partial \hat{\mathbf{x}}_0(\mathbf{x}_t)}{\partial \mathbf{x}_t} \right)^T \right]
\end{aligned}$$

$$
\begin{aligned}
&\overset{(c)}{=} \cancel{\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(0,\mathbf{I})}\left[-\frac{1}{r_t^2}p_0(\mathbf{y}|\hat{\mathbf{x}}_0)\left(\boldsymbol{\xi}^T\frac{\partial\hat{\mathbf{x}}_0(\mathbf{x}_t)}{\partial\mathbf{x}_t}\right)^T\right]} \\
&\quad + \mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(0,\mathbf{I})}\left[-\frac{1}{r_t}\nabla^T_{\mathbf{x}_0}p_0(\mathbf{y}|\hat{\mathbf{x}}_0)\boldsymbol{\xi}\cdot\left(\boldsymbol{\xi}^T\frac{\partial\hat{\mathbf{x}}_0(\mathbf{x}_t)}{\partial\mathbf{x}_t}\right)^T\right] \\
&\overset{(d)}{=}\frac{1}{2\sigma_{\mathbf{y}}^2 r_t}p_0(\mathbf{y}|\mathbf{x}_0)\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(0,\mathbf{I})}\left[(\nabla_{\mathbf{x_0}}\ell(\mathbf{y},\mathcal{A}(\mathbf{x}_0))^T\boldsymbol{\xi})\cdot\left(\boldsymbol{\xi}^T\frac{\partial\hat{\mathbf{x}}_0(\mathbf{x}_t)}{\partial\mathbf{x}_t}\right)^T\right] \\
&=\frac{1}{2\sigma_{\mathbf{y}}^2 r_t}p_0(\mathbf{y}|\mathbf{x}_0)\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(0,\mathbf{I})}\left[(\boldsymbol{\xi}^T\nabla_{\mathbf{x_0}}\ell(\mathbf{y},\mathcal{A}(\mathbf{x}_0))\cdot\left((\frac{\partial\hat{\mathbf{x}}_0(\mathbf{x}_t)}{\partial\mathbf{x}_t})^T\boldsymbol{\xi}\right)\right] \\
&=\frac{1}{2\sigma_{\mathbf{y}}^2 r_t}p_0(\mathbf{y}|\mathbf{x}_0)\mathbb{E}_{\boldsymbol{\xi}\sim\mathcal{N}(0,\mathbf{I})}\left[\mathrm{Tr}\left(\boldsymbol{\xi}\boldsymbol{\xi}^T\nabla_{\mathbf{x_0}}\ell(\mathbf{y},\mathcal{A}(\mathbf{x}_0))\cdot(\frac{\partial\hat{\mathbf{x}}_0(\mathbf{x}_t)}{\partial\mathbf{x}_t})^T\right)\right] \\
&=\frac{1}{2\sigma_{\mathbf{y}}^2 r_t}p_0(\mathbf{y}|\mathbf{x}_0)\nabla_{\mathbf{x}_t}\ell(\mathbf{y},\mathcal{A}(\mathbf{x}_0))
\end{aligned}
\tag{16}
$$

where equality (a) is obtained because $q(\mathbf{x}_0|\mathbf{x}_0)$ is a Gaussian distribution; approximation (b) is obtained via the first order Taylor expansion and is accurate when r_t is small; equation (c) is obtained because $\mathbb{E}[\boldsymbol{\xi}] = 0$ and equation (d) is obtained because $p_0(\mathbf{y}|\mathbf{x}_0)$ is a Gaussian distribution with mean $\mathcal{A}(\mathbf{x}_0)$, then denote $\ell(\mathbf{y},\mathcal{A}(\mathbf{x}_0)) = \|\mathbf{y}-\mathcal{A}(\mathbf{x}_0)\|_2^2$ be the reconstruction loss, the gradient $\nabla_{\mathbf{x}_0}p_0(\mathbf{y}|\mathbf{x}_0) = -\frac{1}{2\sigma_{\mathbf{y}}^2}p_0(\mathbf{y}|\mathbf{x}_0)\nabla\ell(\mathbf{y},\mathcal{A}(\mathbf{x}_0))$.

7 Key parameters for experiments

7.1 DPG

For FFHQ dataset:

- Gaussian Noise $\sigma = 0.05$:
  - Inpainting: $N = 5000$, $C = 180$
  - Super-Resolution: $N = 800$, $C = 160$
  - Gaussian Deblurring: $N = 800$, $C = 200$
  - Motion Deblurring: $N = 500$, $C = 200$
- Poisson Noise $\lambda = 1.0$:
  - Inpainting: $N = 5000$, $C = 180$
  - Super-Resolution: $N = 500$, $C = 150$
  - Gaussian Deblurring: $N = 500$, $C = 150$
  - Motion Deblurring: $N = 500$, $C = 150$
- Nonlinear Inversion Task with Gaussian noise $\sigma = 0.05$:
  - Phase Retrieval: $N = 1000$, $C = 200$
  - Non-Linear Deblurring: $N = 500$, $C = 150$

For ImageNet&LSUN-Bedroom dataset:

- Gaussian Noise $\sigma = 0.05$:
  - Inpainting: $N = 5000, C = 250$
  - Super-Resolution: $N = 500$, $C = 160$
  - Gaussian Deblurring: $N = 500$, $C = 200$
  - Motion Deblurring: $N = 500$, $C = 200$
- Poisson Noise $\lambda = 1.0$:
  - Super-Resolution: $N = 500$, $C = 200$
  - Gaussian Deblurring: $N = 500$, $C = 200$
  - Motion Deblurring: $N = 500$, $C = 200$

7.2 Results on the Image Restoration Process

Next, we will plot the image restoration process and analyze the effect of better score function estimation. We plot the resonctruction loss evolution of DPG (red) and DPS (blue) method in Fig. 2. Shaded area depicts the confidence interval. The mean and confidence interval are obtained by taking the average of 10 runs. According to Fig. 2, DPG always have a smaller reconstruction error in the

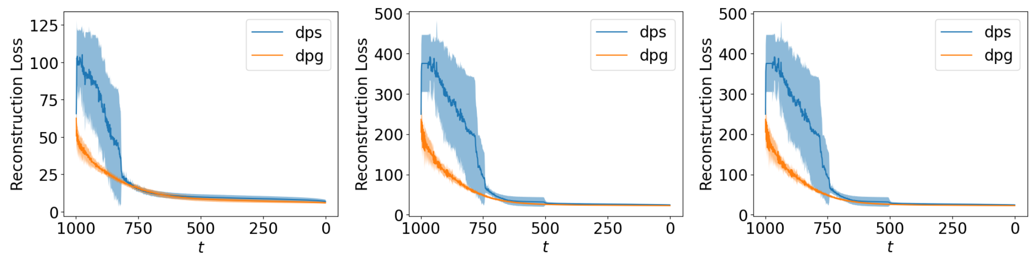


Figure 2: Evolution of the reconstruction error in the image restoration process of super-resolution (left), Gaussian Deblurring (middle) and Motion Deblurring (Right).

earlier stages, and evolution of the reconstruction error is more stable. This leads to the observation that DPG can restore image sketches at an earlier stage, which provides room and opportunity to improve and generation detailed figures in later stage of the diffusion generation process. The image generation results for super-resolution and deblurring tasks are illustrated in Fig. 3.

8 More Experiment Results

8.1 Representative Results on Linear and Non-Linear Inverse Problems

8.2 Experiments with Poisson Noise

We compare the performance of DPG and DPS when the input image $\mathbf{y}$ is distorted by random Poisson noise with rate $\lambda = 1$. Selected Image inverse results are displayed in Fig. 5.

Table 3: Results on Out-of-Distribution Image Inverse Problems on USC-SIPI Dataset

	SR (4×)		**Deblur (Gauss)**		**Deblur (Motion)**	
Method	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑
DPG	**0.245**	22.35	**0.259**	22.35	0.282	22.15
DPS	**0.295**	23.03	0.331	21.17	0.392	18.77
DDRM	0.331	**25.71**	0.417	**23.90**	N/A	N/A

8.3 Results on Out-of-Distribution Inverse Problems

Following Kawar et al. [2022], we test our algorithm on out-of-distribution image inversion problems. We use the unconditional diffusion generation model trained on ImageNet256×256 dataset to solve inverse problems on the USC-SIPI dataset Weber [2006], in which each image does not belong to any ImageNet classes. According to Fig. 6, DPG can successfully solve the inversion problem, and the restored image contains more high frequency details compared with DPG and DDRM.

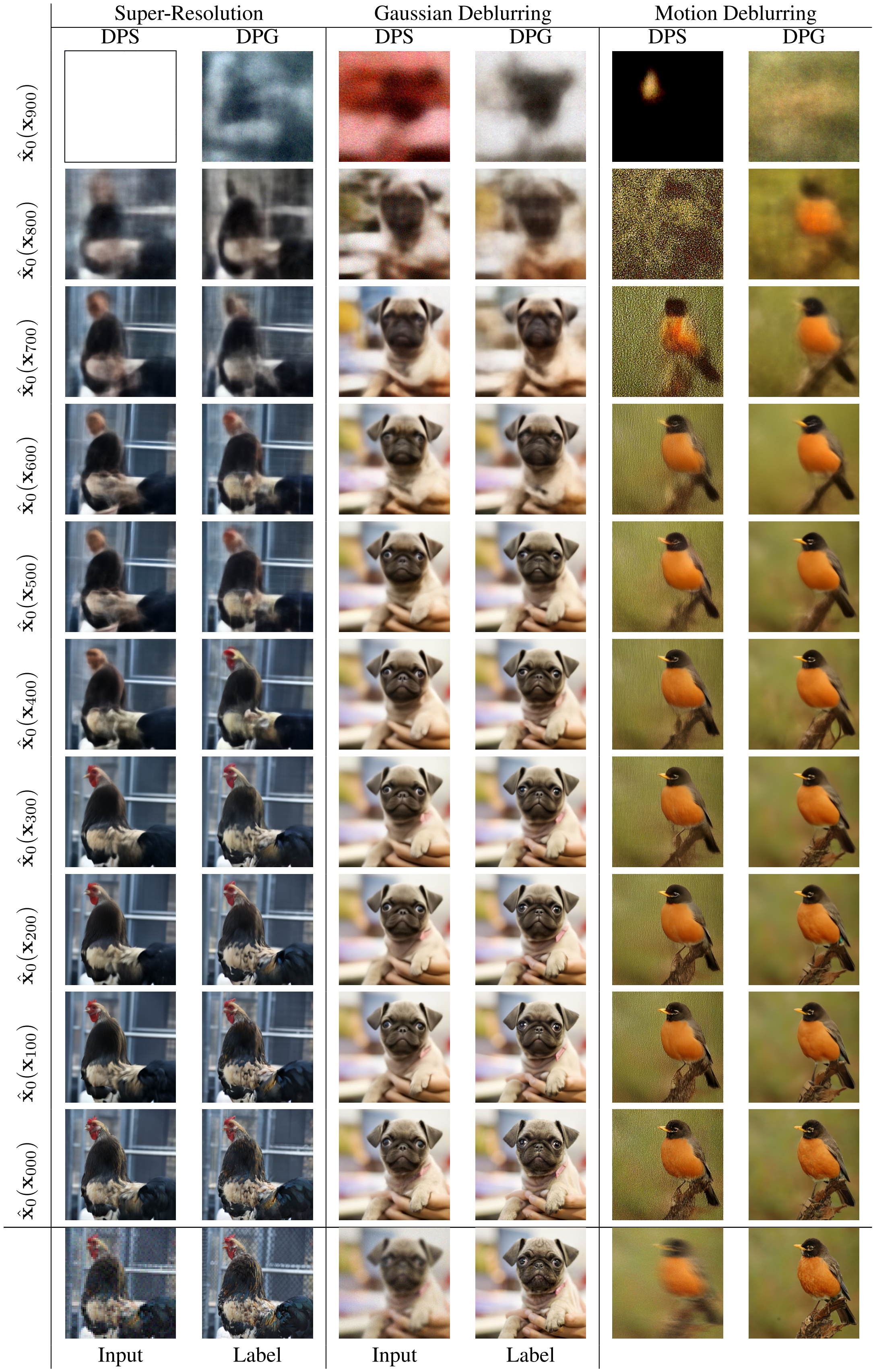
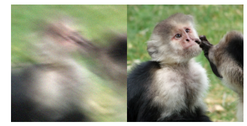


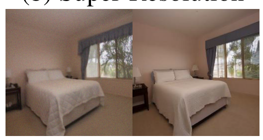
Figure 3: The image restoration process for super-resolution, gaussian deblurring and motion deblurring task.


(a) Inpainting


(b) Super-Resolution


(c) Gaussian Deblurring


(d) Motion Deblurring


(e) Non-linear Deblurring


(f) Uncropping

Figure 4: Examples on solving noisy image inverse problems on ImageNet and LSUN-Bedroom validation set using our proposed method without task specific model finetuning or training.

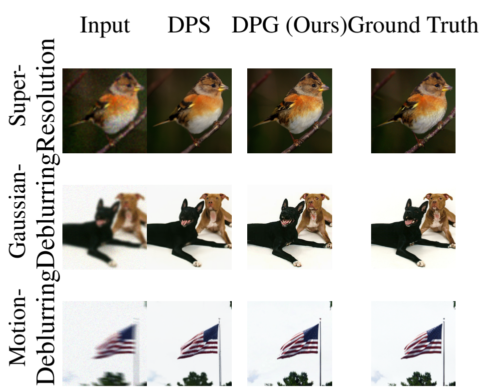


Figure 5: Image Restoration Results on ImageNet with Poisson Noise $\lambda = 1.0$.

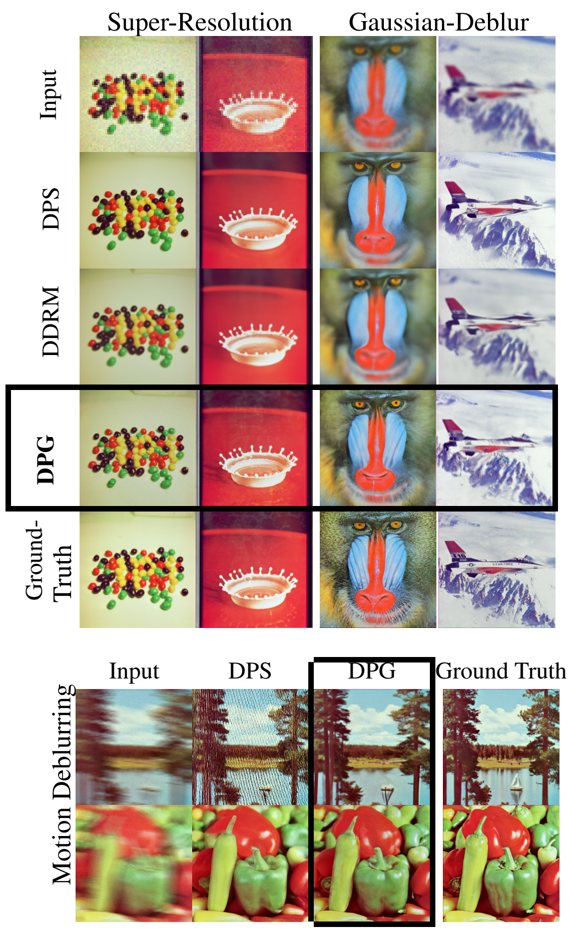


Figure 6: Solving noisy image inversion problems on USC-SIPI Dataset with the pretrained ImageNet Model. Each input has a Gaussian noise $\sigma_{\mathbf{y}} = 0.05$.