Improving Diffusion Models for Inverse Problems using Manifold Constraints

Hyungjin Chung[*,1] **Byeongsu Sim**[*,2] **Dohoon Ryu**[1] **Jong Chul Ye**[3,1,2]
[1] Dept. of Bio and Brain Engineering
[2] Dept. of Mathematical Sciences
[3]Kim Jaechul Graduate School of AI
[*]Equal contribution
Korea Advanced Institute of Science and Technology (KAIST)
{hj.chung, byeongsu.s, dh.ryu, jong.ye}@kaist.ac.kr

Abstract

Recently, diffusion models have been used to solve various inverse problems in an unsupervised manner with appropriate modifications to the sampling process. However, the current solvers, which recursively apply a reverse diffusion step followed by a projection-based measurement consistency step, often produce sub-optimal results. By studying the generative sampling path, here we show that current solvers throw the sample path off the data manifold, and hence the error accumulates. To address this, we propose an additional correction term inspired by the manifold constraint, which can be used synergistically with the previous solvers to make the iterations close to the manifold. The proposed manifold constraint is straightforward to implement within a few lines of code, yet boosts the performance by a surprisingly large margin. With extensive experiments, we show that our method is superior to the previous methods both theoretically and empirically, producing promising results in many applications such as image inpainting, colorization, and sparse-view computed tomography. Code available here

1 Introduction

Diffusion models have shown impressive performance both as generative models themselves [41, 13], and also as unsupervised inverse problem solvers [41, 8, 9, 25] that do not require problem-specific training. Specifically, given a pre-trained unconditional score function (i.e. denoiser), solving the reverse stochastic differential equation (SDE) numerically would amount to sampling from the data generating distribution [41]. For many different inverse problems (e.g. super-resolution [8, 9], inpainting [41, 9], compressed-sensing MRI (CS-MRI) [40, 9], sparse view CT (SV-CT) [40], etc.), it was shown that simple incorporation of the measurement process produces satisfactory conditional samples, even when the model was not trained for the specific problem.

Nevertheless, for certain problems (e.g. inpainting), currently used algorithms often produce unsatisfactory results when implemented naively (e.g. boundary artifacts, as shown in Fig. 1 (b)). The authors in [32] showed that in order to produce high quality reconstructions, one needs to iterate back and forth between the noising and the denoising step at least > 10 times *per iteration*. These iterations are computationally demanding and should be avoided, considering that diffusion models are slow to sample from even without such iterations. On the other hand, a classic result of Tweedie's formula [37, 42] shows that one can perform Bayes optimal denoising in one step, once we know the gradient of the log density. Extending such result, it was recently shown that one can indeed

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

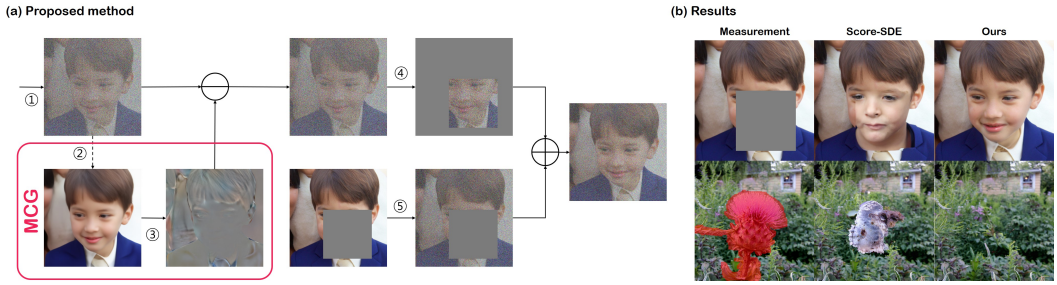


Figure 1: Visual schematic of the MCG correction step. (a) ① Unconditional reverse diffusion generates $\boldsymbol{x}_i$; ② Q_i maps the noisy $\boldsymbol{x}_i$ to generate $\hat{\boldsymbol{x}}_0$; ③ Manifold Constrained Gradient (MCG) $\frac{\partial}{\partial \boldsymbol{x}_i}\|\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{x}}_0)\|_2^2$ is applied to fix the iteration on manifold; ④ Takes the orthogonal complement; ⑤ Samples from $p(\boldsymbol{y}_i|\boldsymbol{y})$, then combines $\boldsymbol{A}\boldsymbol{x}'_{i-1}$ and $\boldsymbol{y}_i$. (b) Representative results of inpainting, compared with score-SDE [41]. Reconstructions with score-SDE produce incoherent results, while our method produces high fidelity solutions.

perform a single-step denoising with learned score functions for denoising problems from the general exponential family [28].

In this work, we leverage the denoising result through Tweedie's formula and show that such denoised samples can be the key to significantly improving the performance of reconstruction using diffusion models across arbitrary linear inverse problems, despite the simplicity in the implementation. Moreover, we theoretically prove that if the score function estimation is globally optimal, the correction term from the manifold constraint enforces the sample path to stay on the plane tangent to the data manifold[1], so by combining with the reverse diffusion step, the solution becomes more stable and accurate.

2 Related Works

2.1 Diffusion Models

Continuous Form For a continuous diffusion process $\boldsymbol{x}(t) \in \mathbb{R}^n$, $t \in [0,1]$, we set $\boldsymbol{x}(0) \sim p_0(\boldsymbol{x}) = p_{data}$, where p_{data} represents the data distribution of interest, and $\boldsymbol{x}(1) \sim p_1(\boldsymbol{x})$, with $p_1(\boldsymbol{x})$ approximating spherical Gaussian distribution, containing no information of data. Here, the forward noising process is defined with the following Itô stochastic differential equation (SDE) [41]:

$$d\boldsymbol{x} = \bar{\boldsymbol{f}}(\boldsymbol{x},t)dt + \bar{g}(t)d\boldsymbol{w}, \tag{1}$$

with $\bar{\boldsymbol{f}} : \mathbb{R}^d \mapsto \mathbb{R}^d$ defining the linear drift function, $\bar{g}(t) : \mathbb{R} \mapsto \mathbb{R}$ defining a scalar diffusion coefficient, and $\boldsymbol{w} \in \mathbb{R}^n$ denoting the standard $n-$dimensional Wiener process. The forward SDE in (1) is coupled with the following reverse SDE by the Anderson's theorem [1, 41]:

$$d\boldsymbol{x} = [\bar{\boldsymbol{f}}(\boldsymbol{x},t) - \bar{g}(t)^2\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})]dt + \bar{g}(t)d\bar{\boldsymbol{w}}, \tag{2}$$

with dt denoting the infinitesimal negative time step, and $\bar{\boldsymbol{w}}$ defining the standard Wiener process running backward in time. Note that the reverse SDE defines the generative process through the score function $\nabla_{\boldsymbol{x}} \log\ p_t(\boldsymbol{x})$, which in practice, is typically replaced with $\nabla_{\boldsymbol{x}} \log p_{0t}(\boldsymbol{x}(t)|\boldsymbol{x}(0))$ to minimize the following denoising score-matching objective

$$\min_{\theta} \mathbb{E}_{t\sim U(\varepsilon,1),\boldsymbol{x}(0)\sim p_0(\boldsymbol{x}),\boldsymbol{x}(t)\sim p_{0t}(\boldsymbol{x}(t)|\boldsymbol{x}(0))} \left[\|\boldsymbol{s}_\theta(\boldsymbol{x}(t),t) - \nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}(t)|\boldsymbol{x}(0))\|_2^2\right]. \tag{3}$$

Once the parameter θ^* for the score function is estimated, one can replace the score function in (2) with $s_{\theta^*}(\boldsymbol{x}(t),t)$ to solve the reverse SDE [41].

Discrete Form Due to the linearity of $\bar{\boldsymbol{f}}$ and $\bar{g}$, the forward diffusion step can be implemented with a simple reparameterization trick [29]. Namely, the general form of the forward diffusion is

$$\boldsymbol{x}_i = a_i\boldsymbol{x}_0 + b_i\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0,\boldsymbol{I}), \tag{4}$$

[1]We coin our method **M**anifold **C**onstrained **G**radient (MCG).

where we have replaced the continuous index $t \in [0, 1]$ with the discrete index $i \in \mathbb{N}$. On the other hand, the discrete reverse diffusion step can in general be represented as

$$\boldsymbol{x}_{i-1} = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{s}_{\theta^*}) + g(\boldsymbol{x}_i)\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}), \tag{5}$$

where we have replaced the ground truth score function with the trained one. We detail the choice of $a_i, b_i, \boldsymbol{f}, g$ in Appendix. B.

2.2 Conditional Generative models for Inverse problems

The main problem of our interest in this paper is the inverse problem, retrieving the unknown $\boldsymbol{x} \in \mathbb{R}^n$ from a measurement $\boldsymbol{y}$:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\epsilon}, \quad \boldsymbol{y} \in \mathbb{R}^m, \boldsymbol{H} \in \mathbb{R}^{m \times n}, \tag{6}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^m$ is the noise in the measurement. Accordingly, for the case of the inverse problems, our goal is to generate samples from a conditional distribution with respect to the measurement $\boldsymbol{y}$, i.e. $p(\boldsymbol{x}|\boldsymbol{y})$. Accordingly, the score function $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ in (2) should be replaced by the conditional score $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}|\boldsymbol{y})$. Unfortunately, this strictly restricts the generalization capability of the neural network since the conditional score should be retrained whenever the conditions change. To address this, recent conditional diffusion models [22, 41, 8, 9] utilize the unconditional score function $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ but rely on a projection-based measurement constraint to impose the conditions. Specifically, one can apply the following:

$$\boldsymbol{x}'_{i-1} = \boldsymbol{f}(\boldsymbol{x}_i, s_\theta) + g(\boldsymbol{x}_i)\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}), \tag{7}$$

$$\boldsymbol{x}_{i-1} = \boldsymbol{A}\boldsymbol{x}'_{i-1} + \boldsymbol{b}_i, \tag{8}$$

where $\boldsymbol{A}, \boldsymbol{b}_i$ are functions of $\boldsymbol{H}, \boldsymbol{y}$, and $\boldsymbol{x}_0$. Note that (7) is identical to the unconditional reverse diffusion step in (5), whereas (8) effectively imposes the condition. It was shown in [9] that any general contraction mapping (e.g. projection onto convex sets, gradient step) may be utilized as (8) to impose the constraint.

Another recent work [25] advancing [26] establishes the state-of-the-art (SOTA) in solving *noisy* inverse problems with unconditional diffusion models, by running the conditional reverse diffusion process in the spectral domain achieved by performing singular value decomposition (SVD), and leveraging approximate gradient of the log likelihood term in the spectral space. The authors show that feasible solutions can be obtained with as small as 20 diffusion steps.

Prior to the development of diffusion models, Plug-and-Play (PnP) models [47, 53, 44] were used in a similar fashion by utilizing a general-purpose unconditional denoiser in the place of proximal mappings in model-based iterative reconstruction methods [5, 3]. Similarly, outside the context of diffusion models, iterative denoising followed by projection-based data consistency was proposed in [44]. In such view, diffusion models can be understood as generative variant of PnPs trained with multiple scales of noise.

GAN-based solvers are also widly explored [4, 10, 20], where the pre-trained generators are tuned at the test time by optimizing over the latent, the parameters, or jointly.

2.3 Tweedie's formula for denoising

In the case of Gaussian noise, a classic result of Tweedie's formula [37] tells us that one can achieve the denoised result by computing the posterior expectation:

$$\mathbb{E}[\boldsymbol{x}|\tilde{\boldsymbol{x}}] = \tilde{\boldsymbol{x}} + \sigma^2 \nabla_{\tilde{\boldsymbol{x}}} \log p(\tilde{\boldsymbol{x}}), \tag{9}$$

where the noise is modeled by $\tilde{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{x}, \sigma^2 I)$. If we consider a diffusion model in which the forward step is modeled as $\boldsymbol{x}_i \sim \mathcal{N}(a_i \boldsymbol{x}_0, b_i^2 I)$ (discrete form), the Tweedie's formula can be rewritten as:

$$\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_i] = (\boldsymbol{x}_i + b_i^2 \nabla_{\boldsymbol{x}_i} \log p(\boldsymbol{x}_i))/a_i. \tag{10}$$

Tweedie's formula is in fact not only relevant to Gaussian denoising in the Bayesian framework, but have also been extended to be in close relation with kernel regression [34]. Moreover, it was shown that it can be applied to arbitrary exponential noise distributions beyond Gaussian [14, 28]. In the following, we use this key property to develop our algorithm.

3 Conditional Diffusion using Manifold Constraints

Although our original motivation of using the measurement constraint step in (8) was to utilize the unconditionally trained score function in the reverse diffusion step in (7), there is room for imposing additional constraints while still using the unconditionally trained score function.

Specifically, the Bayes rule $p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})/p(\boldsymbol{y})$ leads to

$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}|\boldsymbol{y}) = \nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} \log p(\boldsymbol{y}|\boldsymbol{x}). \tag{11}$$

Hence, the score function in the reverse SDE in (7) can be replaced by (11), leading to

$$\boldsymbol{x}'_{i-1} = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{s}_\theta) - \alpha \frac{\partial}{\partial \boldsymbol{x}_i} \|\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}_i)\|_2^2 + g(\boldsymbol{x}_i)\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}) \tag{12}$$

where α and $\boldsymbol{W}$ depend on the noise covariance, if the noise $\boldsymbol{\epsilon}$ in (6) is Gaussian.

Now, one of the important contributions of this paper is to reveal that the Bayes optimal denoising step in (10) from the Tweedie's formula leads to a preferred condition both empirically and theoretically. Specifically, we define the set constraint for $\boldsymbol{x}_i$, called the *manifold constrained gradient (MCG)*, so that the gradient of the measurement term stays on the manifold (see Theorem 1):

$$\boldsymbol{x} \in \mathcal{X}_i, \quad \text{where} \quad \mathcal{X}_i = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x} = (\boldsymbol{x} + b_i^2 \boldsymbol{s}_\theta(\boldsymbol{x}, i))/a_i\} \tag{13}$$

To deal with the potential deviation from the measurement consistency, we again impose the data consistency step (8). Putting them together, the discrete reverse diffusion under the additional manifold constraint and the data consistency can be represented by

$$\boldsymbol{x}'_{i-1} = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{s}_\theta) - \alpha \frac{\partial}{\partial \boldsymbol{x}_i} \|\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_0(\boldsymbol{x}_i))\|_2^2 + g(\boldsymbol{x}_i)\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}), \tag{14}$$

$$\boldsymbol{x}_{i-1} = \boldsymbol{A}\boldsymbol{x}'_{i-1} + \boldsymbol{b}. \tag{15}$$

We illustrate our scheme visually in Fig. 1 (a), specifically for the task of image inpainting. The additional step leads to a dramatic performance boost, as can be seen in Fig. 1 (b). Note that while the mapping (10) does not rely on the measurement, our gradient term in (14) incorporates the information of $\boldsymbol{y}$ so that the gradient of the measurement terms stays on the manifold. In the following, we study the theoretical properties of the method. Further algorithmic details and adaptations to each problem that we tackle are presented in Section C.

We note that the authors of [19] proposed a similar gradient method for the application of temporal imputation and super-resolution. When combining (14) with (15), one can arrive at a similar gradient method proposed in [19], and hence our method can be seen as a generalization to arbitrary linear inverse problems. Furthermore, there are vast literature in the context of PnP models that utilize pre-trained denoisers together with gradient of the log-likelihood to solve inverse problems [30, 48, 11]. Among them, [30] is especially relevant to this work since their method relies on modified Langevin diffusion, together with Tweedie's denoising and projections to the measurement subspace.

4 Geometry of Diffusion Models and Manifold Constrained Gradient

In this section, we theoretically support the effectiveness of the proposed algorithm by showing the problematic behavior of the earlier algorithm and how the proposed algorithm resolves the problem. We defer all proofs in the supplementary section. To begin with, we borrow a geometrical viewpoint of the data manifold.

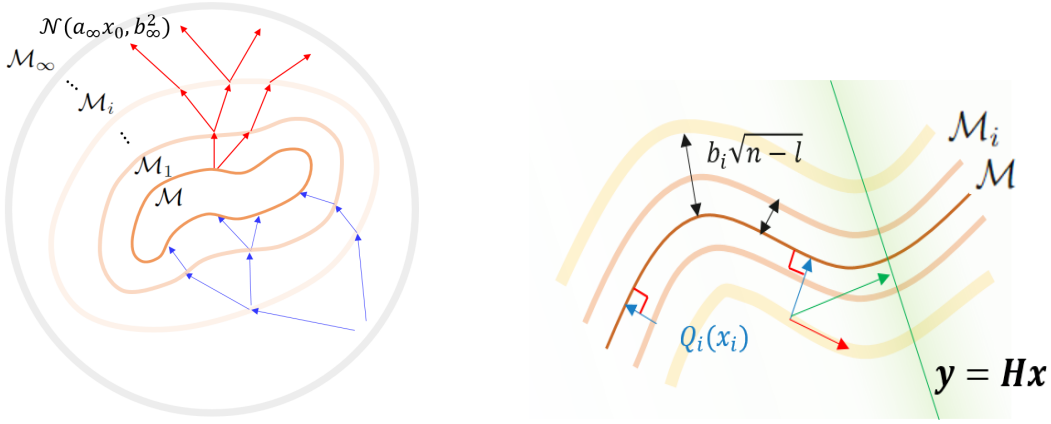
Notation For a scalar a, points $\boldsymbol{x}, \boldsymbol{y}$ and a set A, we use the following notations. $aA := \{a\boldsymbol{x} : \boldsymbol{x} \in A\}$; $d(\boldsymbol{x}, A) := \inf_{\boldsymbol{y} \in A} ||\boldsymbol{x} - \boldsymbol{y}||_2$; $B_r(A) := \{\boldsymbol{x} : d(\boldsymbol{x}, A) < r\}$; $T_{\boldsymbol{x}}\mathcal{M}$: the tangent space to a manifold $\mathcal{M}$ at $\boldsymbol{x}$; $\boldsymbol{J}_f$: the Jacobian matrix of a vector valued function f. We define $p_0 = p_{data}$.

To develop the theory, we need an assumption on the data distribution.

Assumption 1 (Strong manifold assumption: linear structure)**.** *Suppose $\mathcal{M} \subset \mathbb{R}^n$ is the set of all data points, here we call the data manifold. Then, the manifold coincides with the tangent space with dimension $l \ll n$.*

$$\mathcal{M} \cap B_R(\boldsymbol{x}_0) = T_{x_0}\mathcal{M} \cap B_R(x_0) \text{ and } T_{x_0}\mathcal{M} \cong \mathbb{R}^l.$$

Moreover, the data distribution p_0 is the uniform distribution on the data manifold $\mathcal{M}$.



(a) Geometry of diffusion model

(b) MCG correction

Figure 2: In both (a) and (b), the central manifolds represent the data manifold $\mathcal{M}$, encircled by manifolds of noisy data $\mathcal{M}_i$. The concentration on the manifold of noisy data and the distance from the clean data manifold are prescribed by Proposition 1. In (a), the backward (resp. forward) step depicted by blue (resp. red) arrows can be considered as transitions from $\mathcal{M}_i$ to $\mathcal{M}_{i-1}$ (resp. $\mathcal{M}_{i-1}$ to $\mathcal{M}_i$). In (b), arrows refer to the directions of conventional projection onto convex sets (POCS) step (green arrow) and MCG step (red arrow) which can be predicted by Theorem 1.

We need to recall that the conventional manifold assumption is about the intrinsic geometry of data points having a low dimensional nature. However, we assume more in this work: the manifold is locally linear. Although this stronger assumption might narrow the practice of the theory, the geometric approach may provide new insights on diffusion models. Under this assumption, the following proposition shows how the data perturbed by noise lies in the ambient space, illustrated pictorially in Fig. 2a.

Proposition 1 (Concentration of noisy data)**.** *Consider the distribution of noisy data* $p_i(\boldsymbol{x}_i) = \int p(\boldsymbol{x}_i|\boldsymbol{x})p_0(\boldsymbol{x})d\boldsymbol{x}, p(\boldsymbol{x}_i|\boldsymbol{x}) \sim \mathcal{N}(a_i\boldsymbol{x}, b_i^2\boldsymbol{I})$*. Then* $p_i(\boldsymbol{x}_i)$ *is concentrated on* $(n-1)$*-dim manifold* $\mathcal{M}_i := \{\boldsymbol{y} \in \mathbb{R}^n : d(\boldsymbol{y}, a_i\mathcal{M}) = r_i := b_i\sqrt{n-l}\}$*. Rigorously,* $p_i(B_{\epsilon r_i}(\mathcal{M}_i)) > 1-\delta$*, for some small* $\epsilon, \delta > 0$*.*

Remark 1 (Geometric interpretation of the diffusion process)**.** *Considering Proposition 1, the manifolds of noisy data can be interpreted as interpolating manifolds between the two: the hypersphere, where pure noise* $\mathcal{N}(a_\infty\boldsymbol{x}_0, b_\infty^2)$ *is concentrated, and the clean data manifold. In this regard, the diffusion steps are mere transitions from one manifold to another and the diffusion process is a transport from the data manifold to the hypersphere through interpolating manifolds. See Fig. 2a.*

Remark 2. *We can infer from the proposition that the score functions are trained only with the data points concentrated on the noisy data manifolds. Therefore, inaccurate inference might be caused by application of a score function on points away from the noisy data manifold.*

Proposition 2 (score function)**.** *Suppose* s_θ *is the minimizer of the denoising score matching loss in* (3)*. Let* Q_i *be the function that maps* $\boldsymbol{x}_i$ *to* $\hat{\boldsymbol{x}}_0$ *for each* i*,*

$$Q_i : \mathbb{R}^d \to \mathbb{R}^d, \boldsymbol{x}_i \mapsto \hat{\boldsymbol{x}}_0 := \frac{1}{a_i}(\boldsymbol{x}_i + b_i^2 s_\theta(\boldsymbol{x}_i, i)).$$

Then, $Q_i(\boldsymbol{x}_i) \in \mathcal{M}$ *and* $\boldsymbol{J}_{Q_i}^2 = \boldsymbol{J}_{Q_i} = \boldsymbol{J}_{Q_i}^T : \mathbb{R}^d \to T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}$*. Intuitively,* Q_i *is locally an orthogonal projection onto* $\mathcal{M}$*.*

According to the proposition, the score function only concerns the normal direction of the data manifold. In other words, the score function cannot discriminate two data points whose difference is tangent to the manifold. In solving inverse problems, however, we desire to discriminate data points to reconstruct the original signal, and the discrimination is achievable by measurement fidelity. In order to achieve the original signal, the measurement plays a role in correcting the tangent component near the data manifold. Furthermore, with regard to remark 2, diffusion model-based inverse problem solvers should follow the tangent component. The following theorem shows how existing algorithms and the proposed method are different in this regard.

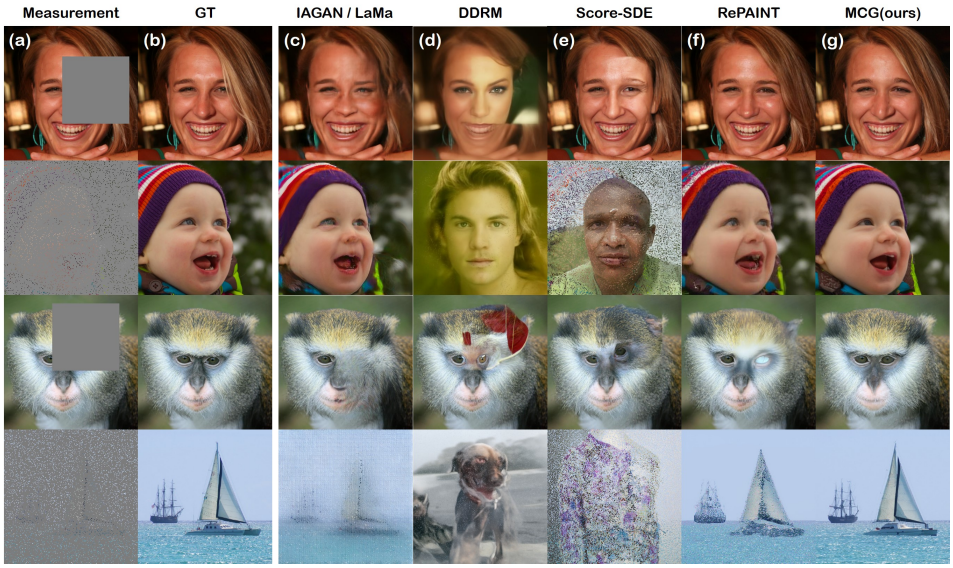


Figure 3: Inpainting results on FFHQ (1st, 2nd row) and ImageNet (3rd, 4th row). (a) Measurement, (b) Ground truth, (c) IAGAN [20] for FFHQ, LaMa [43] for ImageNet, (d) DDRM [25], (e) Score-SDE [41], (f) RePAINT [32], (g) MCG (Ours). Out of 256×256 image, the 1st and the 3rd row is masked with size 128×128 box. 92% of pixels (all RGB channels) from the images in the 2nd and 4th row are blocked.

Theorem 1 (Manifold constrained gradient)**.** *A correction by the manifold constrained gradient does not leave the data manifold. Formally,*

$$\frac{\partial}{\partial \boldsymbol{x}_i}\|\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{x}}_0)\|_2^2 = -2\boldsymbol{J}_{Q_i}^T\boldsymbol{H}^T\boldsymbol{W}^T\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{H}\hat{\boldsymbol{x}}_0) \in T_{\hat{\boldsymbol{x}}_0}\mathcal{M},$$

the gradient is the projection of the data fidelity term onto $T_{\hat{\boldsymbol{x}}_0}\mathcal{M}$,

This theorem suggests that in diffusion models, the naive measurement fidelity step (without considering the data manifold) pushes the inference path out of the manifolds and might lead to inaccurate reconstruction. (To see this pictorially, see section. D, and Fig. 7.) On the other hand, our correction term from the manifold constraint guides the diffusion to lie on the data manifold, leading to better reconstruction. Such geometric views are illustrated in Fig. 2b.

Remark 3. *One may concern that the suboptimality of the denoising score matching loss optimization may lead to inaccurate inference of the MCG steps. In practice, however, most of the error in denoising score matching is concentrated on $t \sim 1$[9], and in such region, the Tweedie's inference cannot make meaningful images. That is, the score function cannot detect the data manifold. Nonetheless, in this regime, the magnitudes of the MCGs are small when the denoising score is inaccurate, and hence the matters arising from suboptimality is minimal. As $t \to 0$, the estimation becomes exact, and subsequently leads to accurate implementation of the MCG.*

5 Experiments

For all tasks, we aim to verify the superiority of our method against other diffusion model-based approaches, and also against strong supervised learning-based baselines. Further details can be found in Section. F.

Datasets and Implementation For inpainting, we use FFHQ 256×256 [24], and ImageNet 256×256 [12] to validate our method. We utilize pre-trained models from the open sourced repository based on the implementation of ADM (VP-SDE) [13]. We validate the performance on 1000 held-out validation set images for both FFHQ and ImageNet dataset. For the colorization task, we use FFHQ 256×256, and LSUN-bedroom 256×256 [51]. We use pre-trained score functions from

	FFHQ (256 × 256)								ImageNet (256 × 256)					
	Box		Random		Extreme		Wide masks		Box		Random		Wide masks	
Method	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
MCG (ours)	**23.7**	0.089	**21.4**	**0.186**	**30.6**	**0.366**	**22.1**	0.099	**25.4**	0.157	**34.8**	**0.308**	21.9	0.148
Score-SDE [41]	30.3	0.135	109.3	0.674	48.6	0.488	29.8	0.132	43.5	0.199	143.5	0.758	25.9	0.150
RePAINT* [32]	25.7	0.093	38.1	0.240	35.9	0.398	24.2	0.108	26.1	0.156	59.3	0.387	37.0	0.205
DDRM [25]	28.4	0.109	111.6	0.774	48.1	0.532	27.5	0.113	88.8	0.386	99.6	0.767	80.6	0.398
LaMa [43]	27.7	**0.086**	188.7	0.648	61.7	0.492	23.2	**0.096**	26.8	**0.139**	134.1	0.567	**20.4**	**0.140**
AOT-GAN [52]	29.2	0.108	97.2	0.514	69.5	0.452	28.3	0.106	35.3	0.163	119.6	0.583	29.8	0.161
ICT [49]	27.3	0.103	91.3	0.445	56.7	0.425	26.9	0.104	31.9	0.148	131.4	0.584	25.4	0.148
DSI [35]	27.9	0.096	126.4	0.601	77.5	0.463	28.3	0.102	34.5	0.155	132.9	0.549	24.3	0.154
IAGAN [20]	26.3	0.098	41.5	0.279	56.1	0.417	23.8	0.110	-	-	-	-	-	-

Table 1: Quantitative evaluation (FID, LPIPS) of inpainting task on FFHQ and ImageNet. *: Re-implemented with our score function. MCG, Score-SDE, RePAINT, and DDRM all share the same score function and differ only in the inference method. **Bold**: Best, under: second best.

score-SDE [41] based on VE-SDE. We use 300 validation images for testing the performance with respect to the LSUN-bedroom dataset. For experiments with CT, we train our model based on `ncsnpp` as a VE-SDE from score-SDE [41], on the 2016 American Association of Physicists in Medicine (AAPM) grand challenge dataset, and we process the data as in [23]. Specifically, the dataset contains 3839 training images resized to 256×256 resolution. We simulate the CT measurement process with parallel beam geometry with evenly-spaced 180 degrees. Evaluation is performed on 421 held-out validation images from the AAPM challenge.

Inpainting Score-SDE [41], REPAINT [32], DDRM [25] were chosen as baseline diffusion models to compare against the proposed method. For a fair comparison, we use the same score function for all methods including MCG, and only differentiate the inference method that is used. Another class of generative models: GAN-based inverse problem solver, IAGAN [20] is considered as a comparison method for FFHQ specifically. We also include comparisons against supervised learning based baselines: LaMa [43], AOT-GAN [52], ICT [49], and DSI [35]. We use various forms of inpainting masks: box (128 × 128 sized square region is missing[2]), extreme (only the box region is existent), random (90-95% of pixels are missing), and LaMa-wide. Quantitative evaluation is performed with two metrics - Frechet Inception Distance (FID)-1k [17], and Learned Perceptual Image Patch Similarity (LPIPS) [54].

Data	FFHQ(256×256)		LSUN(256×256)	
Method	SSIM ↑	LPIPS ↓	SSIM ↓	LPIPS ↓
MCG (ours)	0.951	**0.146**	**0.959**	**0.160**
Score-SDE [41]	0.936	0.180	0.945	0.199
DDRM [25]	0.948	0.154	0.957	0.182
cINN [2]	**0.952**	0.166	0.952	0.180
pix2pix [21]	0.935	0.184	0.947	0.174

Table 2: Quantitative evaluation (SSIM, LPIPS) of colorization task. **Bold**: best, under: second best.

Our method outperforms the diffusion model baselines [41, 32, 25] by a large margin. Moreover, our method is also competitive with, or even better than the best-in-class fully supervised methods, as can be seen in Table 1. In Fig. 3, we depict representative results that show the superiority of the method, where we see in both the box-type and random dropping that MCG performs very well on all experiments.

	AAPM (256 × 256)			
Views	18		30	
Method	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
MCG (ours)	**33.57**	**0.956**	**36.09**	**0.971**
Score-CT [40]	29.85	0.897	31.97	0.913
SIN-4c-PRN [50]	26.96	0.850	30.23	0.917
cGAN [15]	24.38	0.823	27.45	0.927
FISTA-TV [3]	21.57	0.791	23.92	0.861

Table 3: Quantitative evaluation (PSNR, SSIM) of CT reconstruction task. **Bold**: best.

Colorization We choose score-SDE [41], and DDRM [25] as diffusion-model based comparison methods, and also compare against cINN [2], and pix2pix [21]. Two metrics were used for evaluation: structural similarity index (SSIM), and LPIPS. Consistent with the findings from inpainting, we achieve much improved performance than score-SDE, and also is favorable against state-of-the-art (SOTA) superivsed learning based methods. In Table 2, we see that the proposed

[2]The location of the box is sampled uniformly within 16 pixel margin of each side.

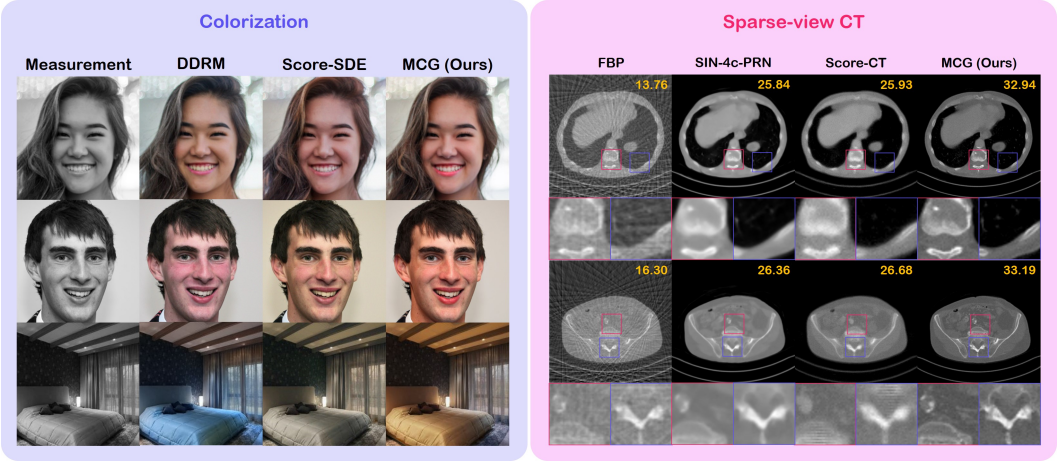


Figure 4: Colorization results on FFHQ / LSUN-bedroom, Sparse view CT reconstruction results on AAPM.

method outperforms all other methods in terms of both PSNR/LPIPS in LSUN-bedroom, and also achieves strong performance in the colorization of FFHQ dataset.

CT reconstruction To the best of our knowledge, [40] is the only method that tackles CT reconstruction directly with diffusion models. We compare our method against [40], which we refer to as score-CT henceforth. We also compare with the best-in-class supervised learning methods, cGAN [15] and SIN-4c-PRN [50]. As a compressed sensing baseline, FISTA-TV [3] was included, along with the analytical reconstruction method, FBP. We use two standard metrics - peak-signal-to-noise-ratio (PSNR), and SSIM for quantitative evaluation. From Table 3, we see that the newly proposed MCG method outperforms the previous score-CT [40] by a large margin. We can observe the superiority of MCG over other methods more clearly in Fig. 4, where MCG reconstructs the measurement with high fidelity and detail. All other methods including the fully supervised baselines fall behind the proposed method.

Ablation studies We perform three ablation studies: 1) As both the MCG term and the projection term contain information about the measurement $\boldsymbol{y}$, we observe the contribution of each term to the fixed solution. To further clarify the efficacy of the gradient step combined with Tweedie's denoising, we also consider the case where the gradient of the log likelihood is computed not in the noiseless regime, but in the noise level matching the current iteration. Specifically, we define $\boldsymbol{x}'_{i-1} := \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{s}_\theta) + g(\boldsymbol{x}_i)\boldsymbol{z}$, $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I})$, $\boldsymbol{y}_{i-1} \sim p(\boldsymbol{y}_{i-1}|\boldsymbol{y}_0)$, and implement the gradient step as $\nabla_{\boldsymbol{x}_i}\|\boldsymbol{y}_{i-1} - \boldsymbol{H}\boldsymbol{x}'_{i-1}\|_2^2$. 2) As the performance of diffusion models depend heavily on the number of NFEs, we observe the trade-off of each diffusion model when varying the NFE from 20 to 1000. Moreover, for completeness, we measure the runtime of each algorithms including the non-diffusion based methods in wall-clock time computed with a commodity GPU in Table. 4. 3) Setting $\alpha = 0.0$ reduces our method to [9]. We show the difference in the performance by varying the values of α.

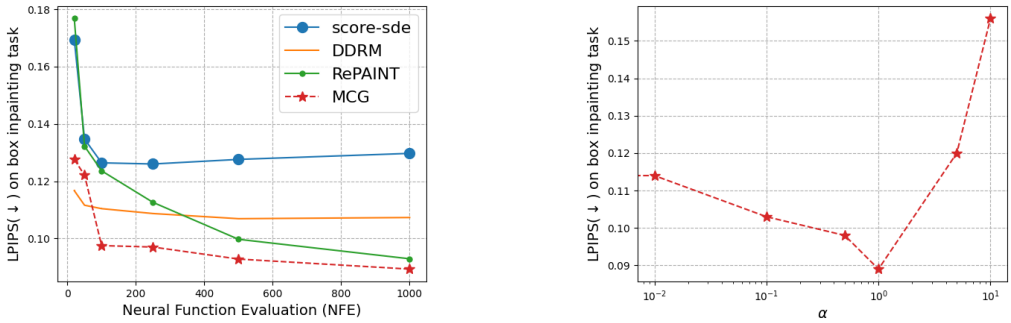
Method	Wall-clock time [s]
Score-SDE [41]	38.68
RePAINT [32]	247.6
DDRM [25]	2.117
LaMa [43]	0.629
AOT-GAN [52]	0.082
ICT [49]	144.6
DSI [35]	36.64
IAGAN [20]	518.47
Ours	81.59

Table 4: Runtime for each algorithm in Wall-clock time: Computed with a single GTX 1080Ti GPU.

First, we see in Table. 5 that using only the MCG step leads to improved performance in terms of LPIPS, but introduces error in the measurement consistency (measured with MSE). Combining both the projection and MCG leads to perfect data consistency along with further improved

Method	LPIPS(↓)	MSE(MC)
Proj.	0.138	0
$\nabla_{\boldsymbol{x}_i}\|\boldsymbol{y}_{i-1} - \boldsymbol{H}\boldsymbol{x}'_{i-1}\|_2^2$	0.271	12.99
$\nabla_{\boldsymbol{x}_i}\|\boldsymbol{y}_{i-1} - \boldsymbol{H}\boldsymbol{x}'_{i-1}\|_2^2$ + Proj.	0.128	0
$\nabla_{\boldsymbol{x}_i}\|\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_0\|_2^2$	0.124	10.7
$\nabla_{\boldsymbol{x}_i}\|\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_0\|_2^2$ + Proj. (**Ours**)	**0.089**	**0**

Table 5: LPIPS & Measurement consistency (MC) vs. method



(a) LPIPS vs. NFE

(b) LPIPS vs. α'

Figure 5: Ablation studies performed with box inpainting task on FFHQ 256×256 data.

reconstruction. When considering gradient steps without Tweedie's denoising (i.e. keeping the noise level at the ith step), the performance heavily degrades, especially when implemented without the projection steps. Here, we see that the proposed denoising step to utilize $\hat{\boldsymbol{x}}_0$ is indeed the key to the superior performance.

Second, looking at Fig. 5a, we immediately see that the graph of MCG stays in the lowest (best) LPIPS regime across all NFEs by a large margin, except for when the NFE drops below 100. Here, DDRM [25] takes over the 1st place - allegedly due to the DDIM sampling strategy they take. The performance of RePAINT deteriorates rapidly as we decrease NFE. Furthermore, we observe that the LPIPS of score-SDE [41] actually *increases* (i.e. worsen), as we increase the number of NFEs from a few hundred to one thousand. This suggests that the inference process that score-SDE takes (i.e. projection only) is inherently flawed, and cannot be corrected by taking small enough steps. In Table. 4, we list the runtime of all the methods that were used for comparison in the task of inpainting. Note that the proposed method takes longer for compute than score-SDE albeit having the same NFE. The gap is due to the backpropagation steps that are required for the MCG step, where the gap can be potentially ameliorated by switching to JAX [6] implementation from the current PyTorch implementation.

Lastly, we observe the difference in the performance as we vary the values of α. Implementation-wise, we find that we yield superior results when normalizing the squared norm with the norm of itself (e.g. $\alpha = \alpha'/\|\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_0)\|$, where α' is some constant). In order to avoid cluttered notation, we instead experiment with changing the values of α' in Fig. 5b. Inspecting Fig. 5b, we see that α values within the range $[0.1, 1.0]$ produce satisfactory results. α values that are too low do not fully enjoy the advantages of MCG and collapses to the projection-only method, while using too high values of α results in exploding gradients, and the reconstruction saturates.

Properties of our method Our proposed method is fully unsupervised and is not trained on solving a specific inverse problem. For example, our box masks and random masks have very different forms of erasing the pixel values. Nevertheless, our method generalizes perfectly well to such different measurement conditions, while other methods have a large performance gap between the different mask shapes. We further note two appealing properties of our method as an inverse problem solver: 1) the ability to generate multiple solutions given a condition, and 2) the ability to maintain perfect measurement consistency. The former ability often lacks in supervised learning-based methods [43, 50], and the latter is often not satisfied for some unsupervised GAN-based solutions [10, 4].

6 Conclusion

In this work, we proposed a general framework that can greatly enhance the performance of the diffusion model-based solvers for solving inverse problems. We showed several promising applications - inpainting, colorization, sparse view CT reconstruction, and showed that our method can outperform the current state-of-the-art methods. We analyzed our method theoretically and show that

MCG prevents the data generation process from falling off the manifold, thereby reducing the errors that might accumulate at every step. Further, we showed that MCG controls the direction tangent to the data manifold, whereas the score function controls the direction that is normal, such that the two components complement each other.

Limitations and Broader Impact The proposed method is inherently stochastic since the diffusion model is the main workhorse of the algorithm. When the dimension m is pushed to low values, at times, our method fails to produce high quality reconstructions, albeit being better than the other methods overall. For extreme cases of inpainting (e.g. Half masks) with the ImageNet model, we often observe artifacts in our reconstruction (e.g. generating perfectly symmetric images), which we discuss in further detail in Sec. E. We note that our method is slow to sample from, inheriting the existing limitations of diffusion models. This would likely benefit from leveraging recent solvers aimed at accelerating the inference speed of diffusion models. In line with the arguments of other generative model-based inverse problem solvers, our method is a solver that relies heavily on the underlying diffusion model, and can thus potentially create malicious content such as deepfakes. Further, the reconstructions could intensify the social bias that is already existent in the training dataset.

Acknowledgments and Disclosure of Funding

This research was supported by Field-oriented Technology Development Project for Customs Administration through National Research Foundation of Korea(NRF) funded by the Ministry of Science & ICT and Korea Customs Service (NRF-2021M3I1A1097938), by the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), which is funded by the Ministry of Health & Welfare, Republic of Korea (grant number: HU21C0222), and by the KAIST Key Research Institute (Interdisciplinary Research Group) Project.

References

[1] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[2] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019.

[3] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[4] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning*, pages 537–546. PMLR, 2017.

[5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[6] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[7] Thorsten M Buzug. Computed tomography. In *Springer handbook of medical technology*, pages 311–342. Springer, 2011.

[8] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: Conditioning method for denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[9] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[10] Giannis Daras, Joseph Dean, Ajil Jalal, and Alexandros G Dimakis. Intermediate layer optimization for inverse problems using deep generative models. In *International Conference on Machine Learning*, 2021.

[11] Valentin De Bortoli, Alain Durmus, Marcelo Pereyra, and Ana Fernandez Vidal. Maximum likelihood estimation of regularization parameters in high-dimensional inverse problems: an empirical bayesian approach. part ii: Theoretical analysis. *SIAM Journal on Imaging Sciences*, 13(4):1990–2028, 2020.

[12] 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.

[13] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

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

[15] Muhammad Usman Ghani and W Clem Karl. Deep learning-based sinogram completion for low-dose ct. In *2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pages 1–5. IEEE, 2018.

[16] Richard Gordon, Robert Bender, and Gabor T Herman. Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography. *Journal of theoretical Biology*, 29(3):471–481, 1970.

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[19] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

[20] Shady Abu Hussein, Tom Tirer, and Raja Giryes. Image-adaptive gan based reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3121–3129, 2020.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[22] Zahra Kadkhodaie and Eero Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In *Advances in Neural Information Processing Systems*, volume 34, pages 13242–13254. Curran Associates, Inc., 2021.

[23] Eunhee Kang, Junhong Min, and Jong Chul Ye. A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction. *Medical physics*, 44(10):e360–e375, 2017.

[24] 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.

[25] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

[26] Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769, 2021.

[27] Daniil Kazantsev, Edoardo Pasca, Martin J Turner, and Philip J Withers. Ccpi-regularisation toolkit for computed tomographic image reconstruction with proximal splitting algorithms. *SoftwareX*, 9:317–323, 2019.

[28] Kwanyoung Kim and Jong Chul Ye. Noise2score: Tweedie's approach to self-supervised image denoising without clean images. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[29] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014.

[30] Rémi Laumont, Valentin De Bortoli, Andrés Almansa, Julie Delon, Alain Durmus, and Marcelo Pereyra. Bayesian imaging using plug & play priors: when langevin meets tweedie. *SIAM Journal on Imaging Sciences*, 15(2):701–737, 2022.

[31] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

[32] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using Denoising Diffusion Probabilistic Models. *arXiv preprint arXiv:2201.09865*, 2022.

[33] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[34] Frank Ong, Peyman Milanfar, and Pascal Getreuer. Local kernels that approximate bayesian regularization and proximal operators. *IEEE Transactions on Image Processing*, 28(6):3007–3019, 2019.

[35] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical VQ-VAE. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10775–10784, 2021.

[36] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

[37] Herbert E Robbins. An empirical bayes approach to statistics. In *Breakthroughs in statistics*, pages 388–394. Springer, 1992.

[38] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR*, 2015.

[40] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022.

[41] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR*, 2021.

[42] Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981.

[43] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2149–2159, 2022.

[44] Tom Tirer and Raja Giryes. Image restoration by iterative denoising and backward projections. *IEEE Transactions on Image Processing*, 28(3):1220–1234, 2018.

[45] Github PK Tool, Nov Sun Mon Tue Wed Thu, and Fri Sat. dkazanc/tomobar.

[46] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[47] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.

[48] Ana Fernandez Vidal, Valentin De Bortoli, Marcelo Pereyra, and Alain Durmus. Maximum likelihood estimation of regularization parameters in high-dimensional inverse problems: An empirical bayesian approach part i: Methodology and experiments. *SIAM Journal on Imaging Sciences*, 13(4):1945–1989, 2020.

[49] Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4692–4701, 2021.

[50] Haoyu Wei, Florian Schiffers, Tobias Würfl, Daming Shen, Daniel Kim, Aggelos K Katsaggelos, and Oliver Cossairt. 2-step sparse-view ct reconstruction with a domain-specific perceptual network. *arXiv preprint arXiv:2012.04743*, 2020.

[51] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, 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.

[52] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Aggregated contextual transformations for high-resolution image inpainting. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[53] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.

[54] 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.

[55] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We discuss the limitations in (6).
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We discuss potential negative impacts in (6).
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] We provide all proofs of results in supplementary material.
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include all code for our experiments in the supplementary material. We will release the code once the paper is published.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Due to our limited resources we do not have time to run multiple sets of experiments.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We have cited the original works that released the datasets.

(b) Did you mention the license of the assets? [No] Licenses are standard and can be found online.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include our implementation as the supplementary material. We will release the code upon publication.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] All datasets used in our work are publicly available.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proofs

First, we remind our notation and the assumption.

Notation For a scalar a, points $\boldsymbol{x}, \boldsymbol{y}$ and a set A, we use the following notations: $aA := \{a\boldsymbol{x} : \boldsymbol{x} \in A\}$; $d(\boldsymbol{x}, A) := \inf_{\boldsymbol{y}\in A} ||\boldsymbol{x} - \boldsymbol{y}||_2$; $B_r(A) := \{\boldsymbol{x} : d(\boldsymbol{x}, A) < r\}$; $T_{\boldsymbol{x}}\mathcal{M}$: the tangent space to a manifold $\mathcal{M}$ at $\boldsymbol{x}$; $\boldsymbol{J}_f$: the jacobian matrix of a vector valued function f.

Assumption 1 (Strong manifold assumption: linear structure)**.** *Suppose $\mathcal{M} \subset \mathbb{R}^n$ is the set of all data points, here we call the data manifold. Then, the manifold coincides with the tangent space with dimension $l \ll n$.*

$$\mathcal{M} \cap B_R(\boldsymbol{x}_0) = T_{x_0}\mathcal{M} \cap B_R(x_0) \text{ and } T_{x_0}\mathcal{M} \cong \mathbb{R}^l.$$

Moreover, the data distribution p_0 is the uniform distribution on the data manifold $\mathcal{M}$.

We state our proofs below.

Proposition 1 (Concentration of noisy data)**.** *Consider the distribution of noisy data $p_i(\boldsymbol{x}_i) = \int p(\boldsymbol{x}_i|\boldsymbol{x})p_0(\boldsymbol{x})d\boldsymbol{x}, p(\boldsymbol{x}_i|\boldsymbol{x}) \sim \mathcal{N}(a_i\boldsymbol{x}, b_i^2\boldsymbol{I})$. Then $p_i(\boldsymbol{x}_i)$ is concentrated on $(n-1)$-dim manifold $\mathcal{M}_i := \{\boldsymbol{y} \in \mathbb{R}^n : d(\boldsymbol{y}, a_i\mathcal{M}) = r_i := b_i\sqrt{n-l}\}$. Rigorously, $p_i(B_{\epsilon r_i}(\mathcal{M}_i)) > 1 - \delta$, for some small $\epsilon, \delta > 0$.*

Proof. Suppose that the data manifold is an l-dimensional linear subspace. By rotation and translation, we safely assume that $\mathcal{M} = \{\boldsymbol{x} \in \mathbb{R}^n : x_{l+1} = x_{l+2} = \cdots = x_n = 0\}$. Then, we can simply write $d(\boldsymbol{x}, \mathcal{M}) = \sqrt{x_{l+1}^2 + \cdots + x_n^2}$, and $\mathcal{M}_i = \{\boldsymbol{x} \in \mathbb{R}^n : x_{l+1}^2 + \cdots + x_n^2 = r_i^2\}$. For a given point $\boldsymbol{x}' = (x_1', x_2', \dots) \in \mathcal{M}$, we consider $p(\boldsymbol{x}|\boldsymbol{x}') \sim \mathcal{N}(a_i\boldsymbol{x}', b_i^2 I)$ and obtain a concentration inequality independent to the choice of $\boldsymbol{x}'$. We need the standard Laurent-Massart bound for a chi-square variable [31]. When X is a chi-square distribution with k degrees of freedom,

$$P[X - k \geq 2\sqrt{kt} + 2t] \leq e^{-t},$$
$$P[X - k \leq -2\sqrt{kt}] \leq e^{-t}.$$

As $\frac{x_{l+1}^2}{b_i^2} + \cdots + \frac{x_n^2}{b_i^2}$ is a chi-square distribution with $n - l$ degrees of freedom, by substituting $t = (n-l)\varepsilon'$ in the above bound,

$$P\left[-2(n-l)\sqrt{\varepsilon'} \leq \frac{x_{l+1}^2}{b_i^2} + \cdots + \frac{x_n^2}{b_i^2} - (n-l) \leq 2(n-l)(\sqrt{\varepsilon'} + \varepsilon')\right]$$
$$= P\left[\sqrt{x_{l+1}^2 + \cdots + x_n^2} \in (\, r_i\sqrt{1 - 2\sqrt{\varepsilon'}},\, r_i\sqrt{1 + 2\sqrt{\varepsilon'} + 2\varepsilon'}\,)\right] \geq 1 - 2e^{-(n-l)\varepsilon'}.$$

Note that the above inequality does not depend on $x_1, \dots x_l$, thus the choice of $\boldsymbol{x}' \in \mathcal{M}$. As a result, by setting $\varepsilon = \min\{1 - \sqrt{1 - 2\sqrt{\varepsilon'}}, \sqrt{1 + 2\sqrt{\varepsilon'} + 2\varepsilon'} - 1\}$ and $\delta = 2e^{-(n-l)\varepsilon'}$,

$$p(\boldsymbol{x} \in B_{\varepsilon r_i}(M_i)|\boldsymbol{x}') > 1 - \delta,$$

thus

$$p_i(\boldsymbol{x} \in B_{\varepsilon r_i}(M_i)) = \int p(\boldsymbol{x} \in B_{\varepsilon r_i}(M_i)|\boldsymbol{x}')p(\boldsymbol{x}')d\boldsymbol{x}' > 1 - \delta.$$

□

Proposition 2 (score function)**.** *Suppose s_θ is the minimizer of the denoising score matching loss in* (3)*. Let Q_i be the function that maps $\boldsymbol{x}_i$ to $\hat{\boldsymbol{x}}_0$ for each i,*

$$Q_i : \mathbb{R}^d \to \mathbb{R}^d, \boldsymbol{x}_i \mapsto \hat{\boldsymbol{x}}_0 := \frac{1}{a_i}(\boldsymbol{x}_i + b_i^2 s_\theta(\boldsymbol{x}_i, i)).$$

Then, $Q_i(\boldsymbol{x}_i) \in \mathcal{M}$ and $\boldsymbol{J}_{Q_i}^2 = \boldsymbol{J}_{Q_i} = \boldsymbol{J}_{Q_i}^T : \mathbb{R}^d \to T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}$. Intuitively, Q_i is locally an orthogonal projection onto $\mathcal{M}$.

Proof. To minimize (3), or equivalently,

$$\int ||s_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{x}_0)||_2^2 p(\boldsymbol{x}_t|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}d\boldsymbol{x}_t dt,$$

By differentiating the objective with respect to $s_\theta(\boldsymbol{x}_t, t)$, we have

$$\begin{aligned}
\int \left(s_\theta(\boldsymbol{x}_t, t) - \frac{a_t\boldsymbol{x} - \boldsymbol{x}_t}{b_t^2} \right) p(\boldsymbol{x}_t|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} &= 0 \\
\int s_\theta(\boldsymbol{x}_t, t)p(\boldsymbol{x}_t)p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x} &= \int \frac{a_t\boldsymbol{x} - \boldsymbol{x}_t}{b_t^2} p(\boldsymbol{x}_t)p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x} \\
s_\theta(\boldsymbol{x}_t, t) \int p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x} &= \int \frac{a_t\boldsymbol{x}}{b_t^2} p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x} - \frac{\boldsymbol{x}_t}{b_t^2} \int p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x} \\
\therefore s_\theta(\boldsymbol{x}_t, t) &= \frac{1}{b_t^2}(-\boldsymbol{x}_t + a_t \int \boldsymbol{x}p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x}) \forall \boldsymbol{x}_t, t,
\end{aligned}$$

where we used $p(\boldsymbol{x}_t|\boldsymbol{x})p(\boldsymbol{x}) = p(\boldsymbol{x}, \boldsymbol{x}_t) = p(\boldsymbol{x}_t)p(\boldsymbol{x}|\boldsymbol{x}_t)$, $p(\boldsymbol{x}_t) > 0$, and $\int p(\boldsymbol{x}|\boldsymbol{x}_t)d\boldsymbol{x} = 1$ in each line. Here, $Q_i(\boldsymbol{x}_i) = \int \boldsymbol{x}p(\boldsymbol{x}|\boldsymbol{x}_i)d\boldsymbol{x}$ is the weighted average vector of points on the data manifold as $p(\boldsymbol{x}|\boldsymbol{x}_i)$ is supported on the data manifold. Combining it with the assumption that the manifold is linear, $Q_i(\boldsymbol{x}_i) \in \mathcal{M}$.

Considering the symmetry of $p(\boldsymbol{x}|\boldsymbol{x}_i)$ about $\boldsymbol{x}_i$, $p(\boldsymbol{x}|\boldsymbol{x}_i)$ is a radial function on $\mathcal{M}$, centering around the nearest point to $\boldsymbol{x}_i$ on $\mathcal{M}$. Hence, $Q_i(\boldsymbol{x}_i)$ shall be the nearest point to $\boldsymbol{x}_i$ of all points on $\mathcal{M}$. Therefore, J_{Q_i} is the orthogonal projection onto $T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}$. Stating more rigorously, let $\boldsymbol{u} = \boldsymbol{u}_t + \boldsymbol{u}_n \in \mathbb{R}^n$ for $\boldsymbol{u}_t \in T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}, \boldsymbol{u}_n \perp T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}$. Then, for a scalar s, $Q_i(\boldsymbol{x}_i + s\boldsymbol{u}) = Q_i(\boldsymbol{x}_i) + s\boldsymbol{u}_t$, as only tangent component to the manifold change the nearest point. By differentiating with respect to s, we obtain $\boldsymbol{J}_{Q_i}\boldsymbol{u} = \boldsymbol{u}_t$, thus $\boldsymbol{J}_{Q_i}^2 = \boldsymbol{J}_{Q_i}$. For another vector $\boldsymbol{v} = \boldsymbol{v}_t + \boldsymbol{v}_n$ with $\boldsymbol{v}_t \in T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}, \boldsymbol{v}_n \perp T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}$,

$$\begin{aligned}
\boldsymbol{v}^T\boldsymbol{J}_{Q_i}\boldsymbol{u} &= (\boldsymbol{v}_t + \boldsymbol{v}_n)^T\boldsymbol{u}_t \\
&= \boldsymbol{v}_t^T\boldsymbol{u}_t \\
&= (\boldsymbol{u}_t + \boldsymbol{u}_n)^T\boldsymbol{v}_t \\
&= \boldsymbol{u}^T\boldsymbol{J}_{Q_i}\boldsymbol{v},
\end{aligned}$$

where we applied $\boldsymbol{v}_n^T\boldsymbol{u}_t = 0 = \boldsymbol{u}_n^T\boldsymbol{v}_t$. Therefore, $\boldsymbol{J}_{Q_i}$ is symmetric, i.e. $\boldsymbol{J}_{Q_i}^T = \boldsymbol{J}_{Q_i}$, which concludes this proof. □

Theorem 1 (Manifold constrained gradient)**.** *A correction by the manifold constrained gradient does not leave the data manifold. Formally,*

$$\frac{\partial}{\partial \boldsymbol{x}_i}\|\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_0)\|_2^2 = -2\boldsymbol{J}_{Q_i}^T\boldsymbol{H}^T\boldsymbol{W}^T\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_0) \in T_{\hat{\boldsymbol{x}}_0}\mathcal{M},$$

the gradient is the projection of the data fidelity term onto $T_{\hat{\boldsymbol{x}}_0}\mathcal{M}$,

Proof.

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{x}_i}\|\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_0)\|_2^2 &= -2\boldsymbol{J}_{\boldsymbol{WHQ}_i}^T\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_0) \\
&= -2\boldsymbol{J}_{Q_i}^T\boldsymbol{H}^T\boldsymbol{W}^T\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_0) \\
&= \boldsymbol{J}_{Q_i}d \in T_{Q_i(\boldsymbol{x}_i)}\mathcal{M}
\end{aligned}$$

where $d = -2\boldsymbol{H}^T\boldsymbol{W}^T\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}}_0)$. The first and second equality is given by the chain rule and the last line is by Proposition 2. □

In Fig 6, we illustrate how the proposed algorithm benefits from mixing the MCG step with the conventional POCS step. Pushing the points to the tangent directions, we expect less deviation from the manifold which is attributed to POCS.

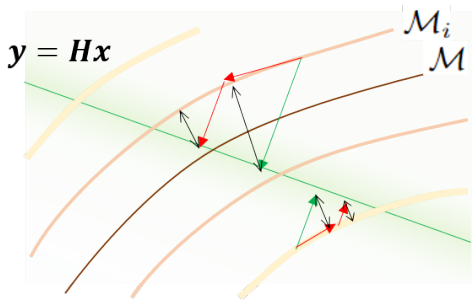


Figure 6: The advantage of mixing the MCG and the POCS steps over the conventional POCS step. Each curve represents a manifold of (noisy) data. Arrows suggest the POCS steps (green arrows) and steps mixing the MCG and the POCS (red arrows). Due to the path along the manifolds, proposed mixing step alleviates reverse diffusion step leaving the manifolds (black arrows).

B Discrete forms of SDE

Here, we review the different types of SDEs and sampling algorithms that we use throughout the paper for completeness. We assume that the time horizon $[0, 1]$ is linearly split up into N discretization segments, such that all intervals have the length $1/N$, if not specified otherwise.

B.1 Forward diffusion

Due to the linearity of the drift and diffusion functions, we can analytically sample from $p(\boldsymbol{x}_i|\boldsymbol{x}_0)$ via reparameterization trick:

$$\boldsymbol{x}_i = a_i \boldsymbol{x}_0 + b_i \boldsymbol{z},\ \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}). \tag{16}$$

In VP-SDE [18], one defines a linearly increasing noise schedule $\beta_1, \beta_2, \ldots, \beta_N \in (0, 1)$. Further, we define $\alpha_i = 1 - \beta_i$, and $\bar{\alpha}_i = \prod_{j=1}^{i} \alpha_j$. Then, the forward diffusion process can be implemented as

$$\boldsymbol{x}_i = \sqrt{\bar{\alpha}_i}\boldsymbol{x}_0 + \sqrt{1 - \bar{\alpha}_i}\boldsymbol{z},\ \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}). \tag{17}$$

In VE-SDE [41], one defines a geometrically increasing noise schedule $\sigma_i = \sigma_0 \left(\frac{\sigma_N}{\sigma_0}\right)^{\frac{i-1}{N-1}}$. Since the drift function is zero, the forward diffusion simply becomes Brownian motion. Concretely,

$$\boldsymbol{x}_i = \boldsymbol{x}_0 + \sigma_i \boldsymbol{z},\ \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}). \tag{18}$$

B.2 Reverse diffusion

First, for the case of VP-SDE, the reverse diffusion step is implemented by

$$\boldsymbol{x}_{i-1} = \frac{1}{\sqrt{\alpha_i}}\left(\boldsymbol{x}_i - \frac{1 - \alpha_i}{\sqrt{1 - \bar{\alpha}_i}}\boldsymbol{z}_\theta(\boldsymbol{x}_i, i)\right) + \sqrt{\tilde{\sigma}_i}\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}), \tag{19}$$

where $\boldsymbol{z}_\theta(\boldsymbol{x}_i, i)$ is trained with the epsilon-matching scheme as in [18], and $\tilde{\sigma}_i$ is set to a learnable parameter as in [13]. Note that eq. (19) was written in terms of $\boldsymbol{z}_\theta(\boldsymbol{x}_i, i)$ and not in terms of the score function, $\boldsymbol{s}_\theta(\boldsymbol{x}_i, i)$. One can re-write the expression using the relation $\boldsymbol{z}_\theta(\boldsymbol{x}_i, i) = -\sqrt{1 - \bar{\alpha}_i}\boldsymbol{s}_\theta(\boldsymbol{x}_i, i)$, as

$$\boldsymbol{x}_{i-1} = \frac{1}{\sqrt{\alpha_i}}\left(\boldsymbol{x}_i + (1 - \alpha_i)\boldsymbol{s}_\theta(\boldsymbol{x}_i, i)\right) + \sqrt{\sigma_i}\boldsymbol{z},\ \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}). \tag{20}$$

Next, for the VE-SDE, the reverse diffusion step using the Euler-Maruyama solver [38] is given as

$$\boldsymbol{x}_{i-1} = \boldsymbol{x}_i + (\sigma_i^2 - \sigma_{i-1}^2)s_\theta(\boldsymbol{x}_i, i) + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\boldsymbol{z},\ \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}). \tag{21}$$

Summary is presented in Table 6.

Type	a_i	b_i	$\boldsymbol{f}(\boldsymbol{x}_i, s_\theta)$	$g(i)$
VP-SDE	$\sqrt{\bar{\alpha}_i}$	$\sqrt{1-\bar{\alpha}_i}$	$\frac{1}{\sqrt{\bar{\alpha}_i}}\left(\boldsymbol{x}_i + (1-\alpha_i)s_\theta(\boldsymbol{x}_i, i)\right)$	$\sqrt{\tilde{\sigma}_i}$
VE-SDE	1	σ_i	$\boldsymbol{x}_i + (\sigma_i^2 - \sigma_{i-1}^2)s_\theta(\boldsymbol{x}_i, i)$	$\sqrt{\sigma_i^2 - \sigma_{i-1}^2}$

Table 6: Choice of $a_i, b_i, \boldsymbol{f}, g$ for each SDE realization.

Algorithm 1 Inpainting (VP, AS)

Require: $\boldsymbol{y}, \boldsymbol{P}, \{\alpha_i\}_{i=1}^N, \{\tilde{\sigma}_i\}_{i=1}^N, s_\theta, \alpha$

$\boldsymbol{x}_N \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ ▷ Initial sampling
for $i = N$ to 1 **do** ▷ Reverse diffusion
  $s \leftarrow s_\theta(\boldsymbol{x}_i, i)$ ▷ Cache score function output
  $\boldsymbol{x}'_{i-1} \leftarrow \frac{1}{\sqrt{\alpha_i}}(\boldsymbol{x}_i + (1-\alpha_i)s)$
  $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  $\boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} + \tilde{\sigma}_i \boldsymbol{z}$ ▷ Unconditional update
  $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  $\hat{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\boldsymbol{x}_i + (1-\bar{\alpha}_i)s)$ ▷ $\hat{\boldsymbol{x}}_0$ prediction
  $\boldsymbol{y}_i \leftarrow \sqrt{\bar{\alpha}_i}\boldsymbol{y} + \sqrt{1-\bar{\alpha}_i}\boldsymbol{z}$
  $\boldsymbol{x}''_{i-1} \leftarrow \boldsymbol{x}'_{i-1} - \alpha \frac{\partial}{\partial \boldsymbol{x}_i}\|\boldsymbol{y} - \boldsymbol{P}\hat{\boldsymbol{x}}_0\|_2^2$ ▷ MCG
  $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  $\boldsymbol{y}_i \leftarrow \sqrt{\bar{\alpha}_i}\boldsymbol{y} + \sqrt{1-\bar{\alpha}_i}\boldsymbol{z}$
  $\boldsymbol{x}_{i-1} \leftarrow (\boldsymbol{I} - \boldsymbol{P}^T\boldsymbol{P})\boldsymbol{x}''_{i-1} + \boldsymbol{P}^T\boldsymbol{y}_i$ ▷ Data consistency
end for
return $\boldsymbol{x}_0$

C Algorithms

Inpainting The forward model for inpainting is given as

$$\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x} + \boldsymbol{\epsilon}, \quad \boldsymbol{P} \in \mathbb{R}^{m \times n}, \tag{22}$$

where $\boldsymbol{P} \in \{0,1\}^{m \times n}$ is the matrix consisting of the columns with standard coordinate vectors indicating the indices of measurement. For the steps in (14), (15), we choose the following

$$\boldsymbol{W} = \boldsymbol{I}, \quad \boldsymbol{A} = \boldsymbol{I} - \boldsymbol{P}^T\boldsymbol{P}, \quad \boldsymbol{b}_i = \boldsymbol{P}^T\boldsymbol{y}_i, \quad \boldsymbol{y}_i \sim q(\boldsymbol{y}_i|\boldsymbol{y}) := \mathcal{N}(\boldsymbol{y}_i | a_i\boldsymbol{y}, b_i^2\boldsymbol{I}). \tag{23}$$

Specifically, $\boldsymbol{A}$ takes the orthogonal complement of $\boldsymbol{x}'_{i-1}$, meaning that the measurement subspace is corrected by $\boldsymbol{y}_i$, while the orthogonal components are updated from $\boldsymbol{x}'_{i-1}$. Note that we use $\boldsymbol{y}_i$ sampled from $\boldsymbol{y}$ to match the noise level of the current estimate.

We provide the algorithm used for inpainting in Algorithm. 1. The sampler is based on basic ancestral sampling (AS) of [18], and the default configuration requires $N = 1000$, $\alpha = 1.0/\|\boldsymbol{y} - \boldsymbol{P}\hat{\boldsymbol{x}}_0\|$ for sampling.

Colorization The forward model for colorization is specified as

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{\epsilon} := \boldsymbol{P}\boldsymbol{M}\boldsymbol{x} + \boldsymbol{\epsilon}, \quad \boldsymbol{P} \in \mathbb{R}^{m \times n}, \quad \boldsymbol{M} \in \mathbb{R}^{n \times n}, \tag{24}$$

where $\boldsymbol{P}$ is the matrix that was used in inpainting, and $\boldsymbol{M}$ is an orthogonal matrix that couples the RGB colormaps[3]. $\boldsymbol{M}^T$ is a matrix that de-couples the channels back to the original space. In other words, one can view colorization as performing imputation in some spectral space. Subsequently, for our colorization method we choose

$$\boldsymbol{W} = \boldsymbol{C}^T, \quad \boldsymbol{A} = \boldsymbol{I} - \boldsymbol{C}^T\boldsymbol{C}, \quad \boldsymbol{b}_i = \boldsymbol{C}^T\boldsymbol{y}_i, \quad \boldsymbol{y}_i \sim q(\boldsymbol{y}_i|\boldsymbol{y}). \tag{25}$$

Again, our forward measurement matrix is orthogonal, and we choose $\boldsymbol{A}$ such that we only affect the orthogonal complement of the measurement subspace.

[3]The matrix $\boldsymbol{M}$ is adopted from the colorization matrix of [41].

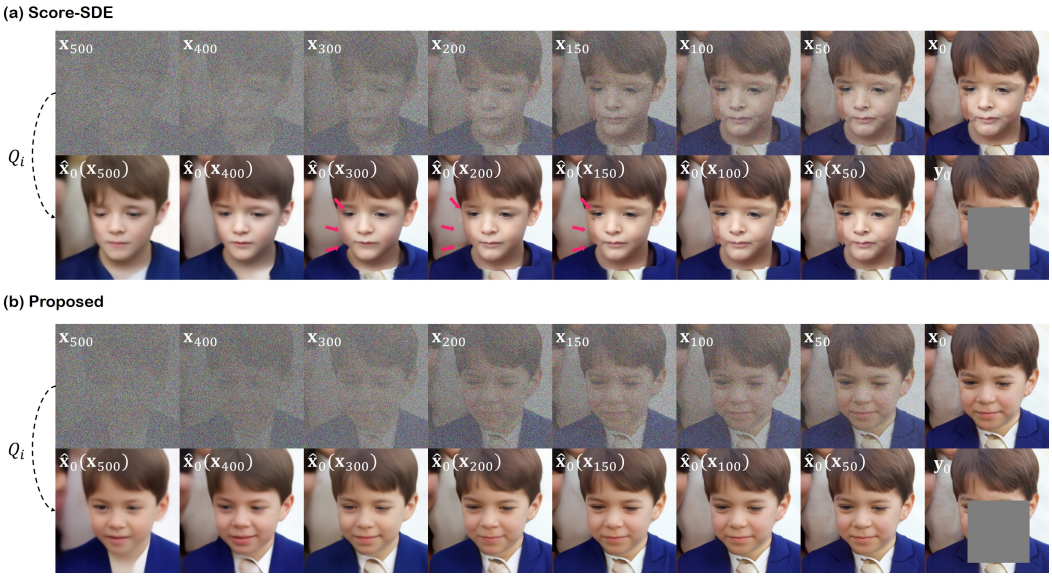


Figure 7: Comparison of the evolution (i.e. generative path) between score-SDE [41], and our method. First rows in (a),(b): Evolution of $\boldsymbol{x}_i$, second rows in (a),(b): Evolution of $\hat{\boldsymbol{x}}_0$.

The sampler for colorization is based on the predictor-corrector (PC) sampler of [41] (VE-SDE), and we choose to apply MCG after every iteration of both predictor, and corrector steps. $N = 2000, \alpha = 0.1/\|\boldsymbol{C}^T(\boldsymbol{y} - \boldsymbol{C}\hat{\boldsymbol{x}}_0)\|$ are chosen as hyper-parameters.

CT Reconstruction For the case of CT reconstruction, the forward model reads

$$\boldsymbol{y} = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{\epsilon}, \quad \boldsymbol{R} \in \mathbb{R}^{m \times n}, \tag{26}$$

where $\boldsymbol{R}$ is the discretized Radon transfrom [7] that measures the projection images from different angles. Note that for CT applications, $\boldsymbol{R}^T$ corresponds to performing backprojection (BP), and $\boldsymbol{R}^\dagger$ corresponds to performing filtered backprojection (FBP). We choose

$$\boldsymbol{W} = \boldsymbol{R}^\dagger, \quad \boldsymbol{A} = \boldsymbol{I} - \boldsymbol{R}^T(\boldsymbol{R}\boldsymbol{R}^T)^\dagger \boldsymbol{R}, \quad \boldsymbol{b}_i = \boldsymbol{R}^T(\boldsymbol{R}\boldsymbol{R}^T)^\dagger \boldsymbol{y}_i, \quad \boldsymbol{y}_i \sim q(\boldsymbol{y}_i|\boldsymbol{y}), \tag{27}$$

where the choice of $\boldsymbol{A}$ reflects that the Radon transform is not orthogonal, and we need the term $(\boldsymbol{R}\boldsymbol{R}^T)^\dagger$ as a term analogous to the filtering step. Indeed, this form of update is known as the algebraic reconstruction technique (ART), a classic technique in the context of CT [16]. We note that this choice is different from what was proposed in [40], where the authors repeatedly apply projection/FBP by explicitly replacing the sinogram in the measured locations. From our experiments, we find that repeated application of FBP is highly numerically unstable, often leading to overflow. This is especially the case when we have limited resources for training data (we use 4k, whereas [40] uses 50k), as we further show in Section 5.

Algorithm for SV-CT reconstruction uses PC sampler (VE-SDE), where we use MCG step after one sweep of corrector-predictor update. We note that this is a design choice, and one may as well use the MCG update step after both the predictor and corrector steps, as was proposed in [41]. We set $N = 2000, \alpha = 0.1/\|\boldsymbol{R}^\dagger(\boldsymbol{y} - \boldsymbol{R}\hat{\boldsymbol{x}}_0)\|$.

D Generative process of the proposed method

In Fig. 7, we depict the comparison of the generative process between the two methods: score-SDE [41], which relies on alternating projections; and our method, which utilizes MCG as correcting steps. In Fig. 7 (a), we can clearly see the unnatural boundary between the masked and the unmasked region forming, and evolving as $t \to 0$, without getting corrected (Visible more clearly in $\hat{\boldsymbol{x}}_0$). On the other hand, thanks to the additional gradient step that *corrects* the errors in the boundary, we see a much more natural evolution of the signal as $t \to 0$ in Fig. 7 (b).

E Limitations

There exists a limitation specifically for the ImageNet dataset when using the proposed algorithm for inpainting. Specifically, as shown in Fig. 8, for the case of half-mask (i.e. the left or right half of the image is zeroed-out), we often see the reconstructions are generated showing symmetries that are unrealistic. Note that this kind of effect is not observed in our FFHQ experiments. Hence, we conjecture that this phenomenon arises from the imperfectness of the learned score function s_θ. Namely, due to the ImageNet dataset being much more diverse and therefore widely known to be a much harder dataset to learn, the subobtimality of the score function may be greater than the FFHQ score function. This could possibly lead to such deficiencies.

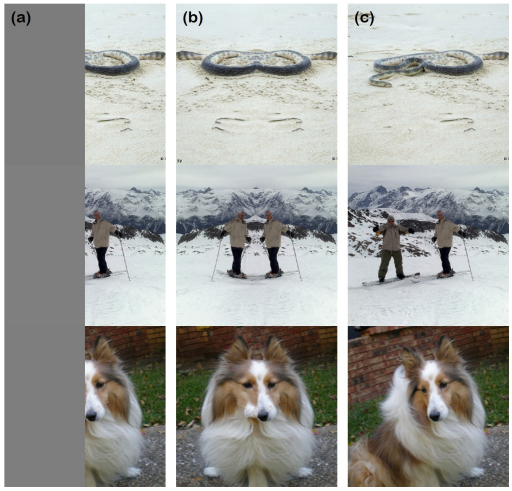


Figure 8: Limitations of the proposed method. (a) Measurement, (b) reconstruction with the proposed method, (c) ground truth.

F Experimental Details

F.1 Implementation details

Training of the score function For inpainting experiments, we take the pre-trained score functions that are available online (FFHQ[4], imagenet[5]). For CT reconstruction experiment, we train a `ncsnpp` model with default configurations as guided in [41] with the VE-SDE framework. The model was trained for 200 epochs with the full training dataset, with a single RTX 3090 GPU. Training took about one week wall-clock time.

Required compute time for inference All our sampling steps detailed in Algorithm C was performed with a single RTX 3090 GPU. The inpainting algorithm based on ADM [13] takes about 90 seconds (1000 NFE) to reconstruct a single image of size 256×256. Our colorization and CT reconstruction algorithm based on score-SDE [41] takes about 600 seconds (4000 NFE) to infer a single 256×256 image.

Code Availability We will open-source our code used in our experiments upon publication to boost reproducibility.

F.2 Comparison methods

F.2.1 Inpainting, Colorization

Score-SDE Score-SDE [41] demonstrated that unconditional diffusion models can be adopted to various inverse problems, such as inpainting and colorization. Our method without the MCG step is identical to score-SDE, and hence we use the same score function, parameters, and sampler as used in the proposed method for reconstruction.

RePAINT RePAINT [32] proposes to iterate between denoising-noising steps multiple times in order to better incorporate inter-dependency between the known and the unknown regions in the case of image inpainting. We use the same score function and sampler for RePAINT as in the proposed method. Following the default configurations in [32], we take $N = 200$ (corresponding to T in [32]), and $U = 10$, where U denotes the count of iterated denoising-noising steps used within a single update index i.

[4] `https://github.com/jychoi118/ilvr_adm`
[5] `https://github.com/openai/guided-diffusion`

DDRM DDRM [25] demonstrates that linear inverse problems can be solved via diffusion models by decomposing the generative process with singular value decomposition (SVD), and performing reverse diffusion sampling in the spectral space. The same score function adopted for the proposed method is used. Using the notations from [25], we choose $\sigma_{\boldsymbol{y}} = 0$, as we are aiming to solve noiseless inverse problem, and $\eta = 0.85, \eta_b = 1$. The number of NFE is set to 20 with the DDRM sampling steps.

LaMa LaMA contains fast Fourier convolution in generator architecture for reconstructing images. We trained the model from scratch using adversarial loss with r1 regularization term with its coefficient 10 and gradient penalty coefficient 0.001. Adam optimizer is used with the fixed learning rate of 0.001 and 0.0001 for discriminator network. For FFHQ and Imagenet dataset, 500k iterations of trainings were done with batch size of 8.

AOT-GAN AOT-GAN consists of a deep image generator with a AOT block which consists of multiple length of residual blocks in parallel. The discriminator is the same architecture with PatchGAN from [55]. We trained the model from the scratch with 0.0001 learning rate using Adam optimizer $\beta_1 = 0$ and $\beta_2 = 0.9$ for both FFHQ and Imagent dataset. 500k iterations of trainings were done with batch size of 8. Also, for style loss and the perceptual loss, VGG19 [39] pretrained on ImageNet [12] was used.

ICT Image completion transformer (ICT) consists of two modules - a transformer model that follows the tokenization procedure to process information in the lower dimensional space, and another guided upsampling module to retrieve the data dimensionality. The encoded features are sampled from a probability distribution via Gibbs sampling, such that one can capture multimodal reconstructions from the same measurement. For both the FFHQ and Imagenet dataset, we used pretrained models provided by the authors.

IAGAN Image adaptive GAN (IAGAN) uses a pre-trained generator and adapts it at test time for the given forward model. Specifically, following compressed sensing using generative model (CSGM) [4], one initializes the latent vector $\boldsymbol{z}$ such that $\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \|y - AG_\theta(\boldsymbol{z})\|$. Then, the latent code and the neural network parameters are jointly optimized through some iterations of $\boldsymbol{z}^{**}, \theta^* = \arg\min_{\boldsymbol{z},\theta} \|y - AG_\theta(\boldsymbol{z})\|$. The final result is achieved by the forward pass through the generator, after which follows the projection into the measurement subspace. For tuning the generator, we follow the default configurations from the official codebase. Since the codebase uses a GAN that generates 1024×1024 images, we downscale the result into 256×256 image as a final post-processing step.

DSI DSI is structured with the combination of VQ-VAE [46], structure generator and texture generator. The architectures were trained separately, with Adam optimizer. When inference, only structure and texture generator was used. We trained the model from scratch. During optimization, the structure generator used linear warm-up schedular and square-root decay schedule used in [36]. We used Adam optimizer on training all models with learning rate of 0.0001 and $\beta_1 = 0.5$ using exponetial moving average (EMA). Training was done for 500k iteration for both FFHQ and Imagenet dataset.

cINN cINN is an invertible neural network which can take in additional conditions as input, and in our case grayscale images. We train the model using default configurations as advised in `https://github.com/VLL-HD/conditional_INNs` without modifications. FFHQ model was trained with the learning rate of 0.0001 for 100 epohcs using the Adam optimizer. LSUN bedroom model was trained with the learning rate of 0.0001 for 30 epochs.

pix2pix Pix2pix is a variant of conditional GAN (cGAN) that takes in as input, the corrupted image. The model is trained in a supervised fashion, with the loss consisting of the reconstruction loss, and the adversarial loss. As the discriminator architecture, we adopt patchGAN [21], and utilize the LSGAN [33] loss, weighting the adversarial loss by the value of 0.1. Similar to cINN, FFHQ model was trained with the learning rate of 0.0001 for 100 epochs using Adam optimizer. LSUN bedroom model was trained with the same configuration for 30 epochs.

F.3 CT reconstruction

Score-CT We use the hyper-parameters as advised in [40] and set $\eta = 0.246, \lambda = 0.841$. The measurement consistency step is imposed after every corrector-predictor sweep as in the proposed method.

SIN-4c-PRN Directly using the official implementation[6] [50], we train the sinogram inpainting network (SIN) with the AAPM dataset for 200 epochs with the batch size of 8, and learning rate of 0.0001. We train two models separately for different number of views - 18, and 30.

cGAN We adopt the implementation of cGAN [15] from SIN-4c-PRN repository [6]. We train the two separate networks for 18 view, and 30 view projection, with the same configuration - 200 epochs, learning rate of 0.0001, and batch size of 8.

FISTA-TV We perform FISTA-TV [3] reconstruction using `TomoBAR` [45], together with the `CCPi` regularization toolkit [27]. Leveraging the default setting, we use the least-squares (LS) data model, and run the FISTA iteration for 300 iterations per image, with the total variation regularization strength set to 0.001.

G Further Experimental Results

We provide extensive set of comparison study for each task in Fig. 9, 11, and 12. Furthermore, in order to illustrate the ability of our method to generate multimodal reconstructions given a measurement, we present further experimental results of inpainting and colorization in the following figures: Fig. 13, 14, 15, and 16

[6] `https://github.com/anonyr7/Sinogram-Inpainting`

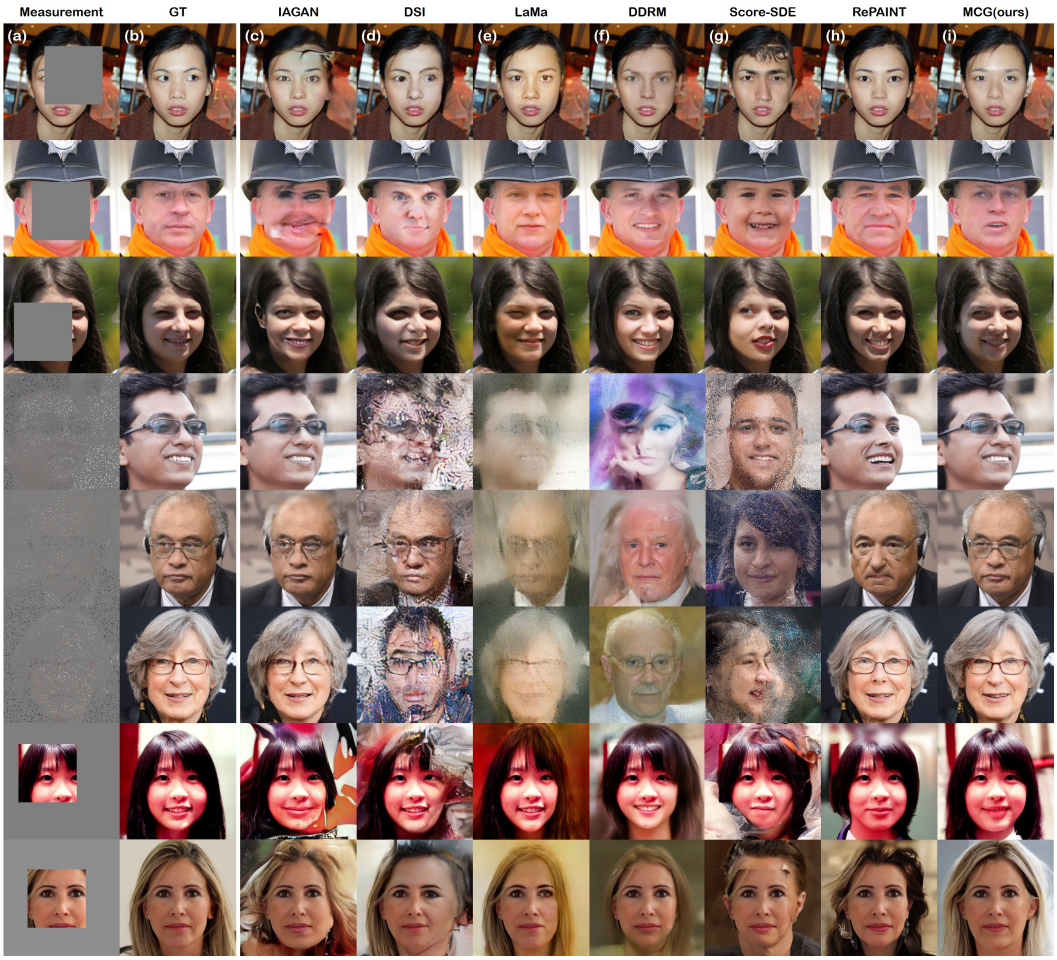


Figure 9: Inpainting results on FFHQ 256×256 data. (a) Measurement, (b) ground truth, (c) IAGAN [20], (d) DSI [35], (e) LaMa [43], (f) DDRM [25], (g) score-SDE [41], (h) RePAINT [32], (i) MCG (ours).

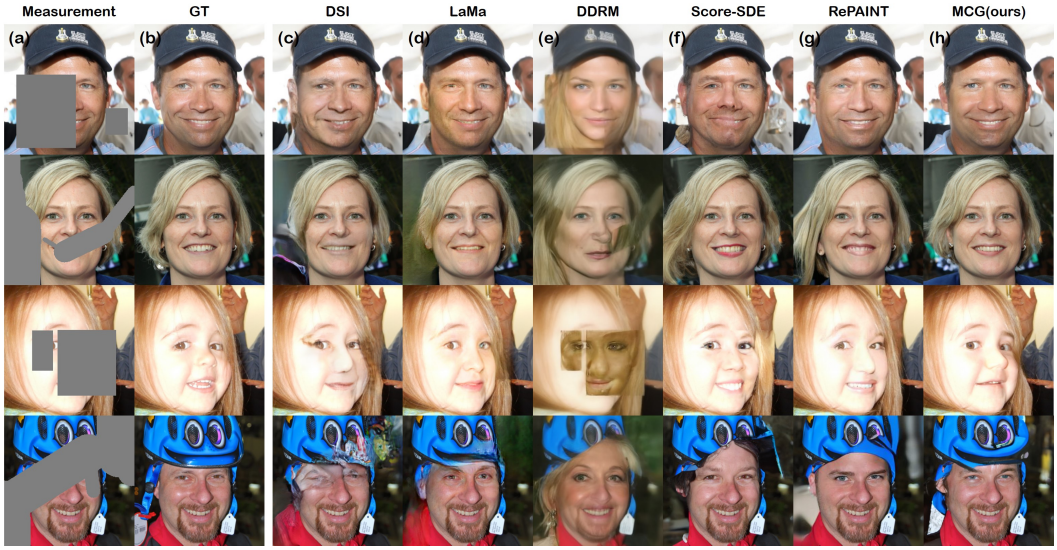


Figure 10: Inpainting results on FFHQ 256×256 data with the LaMa [43] wide mask. (a) Measurement, (b) ground truth, (c) DSI [35], (d) LaMa [43], (e) DDRM [25], (f) score-SDE [41], (g) RePAINT [32], (h) MCG (ours).

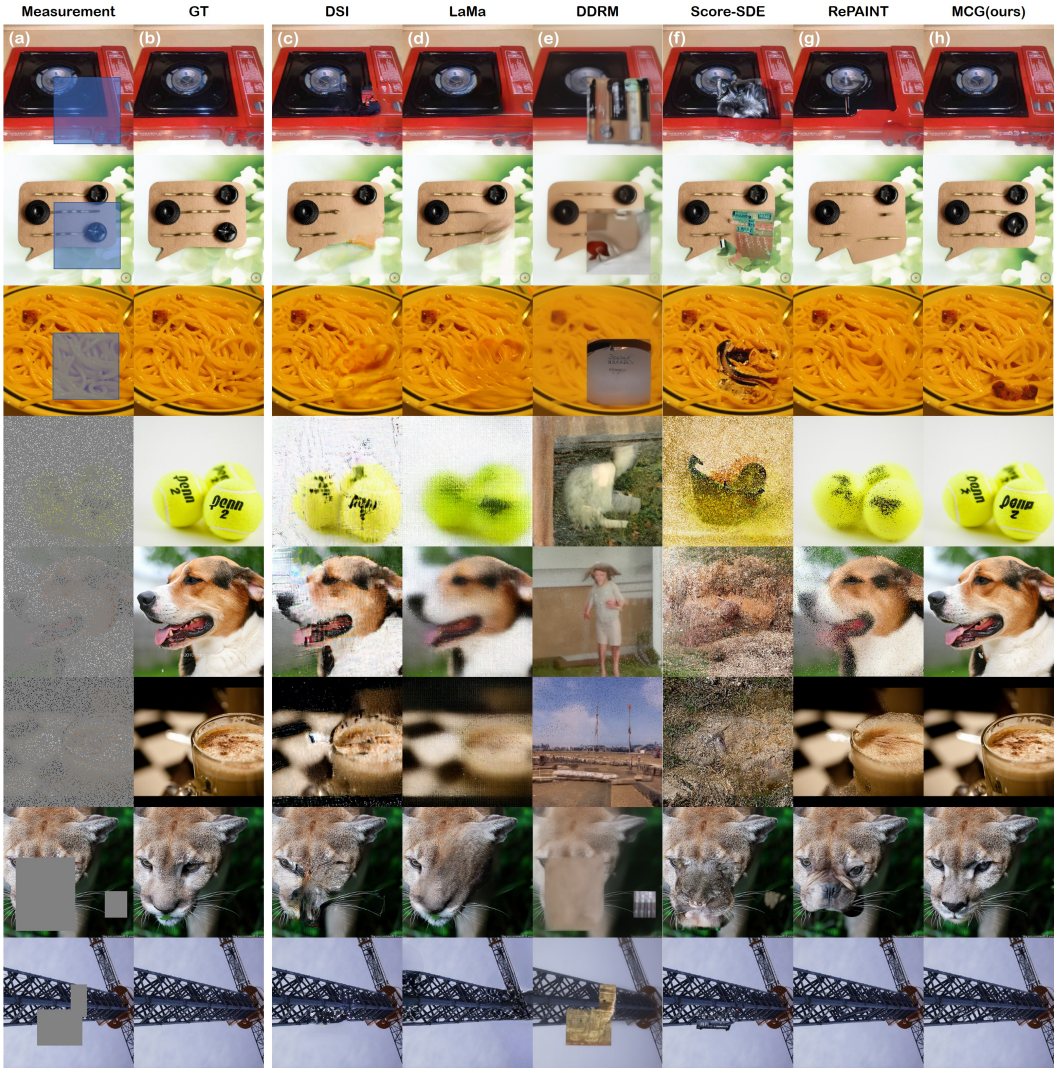


Figure 11: Inpainting results on ImageNet 256×256 data.(a) Measurement, (b) ground truth, (c) DSI [35], (d) LaMa [43], (e) DDRM [25], (f) score-SDE [41], (g) RePAINT [32], (h) MCG (ours).

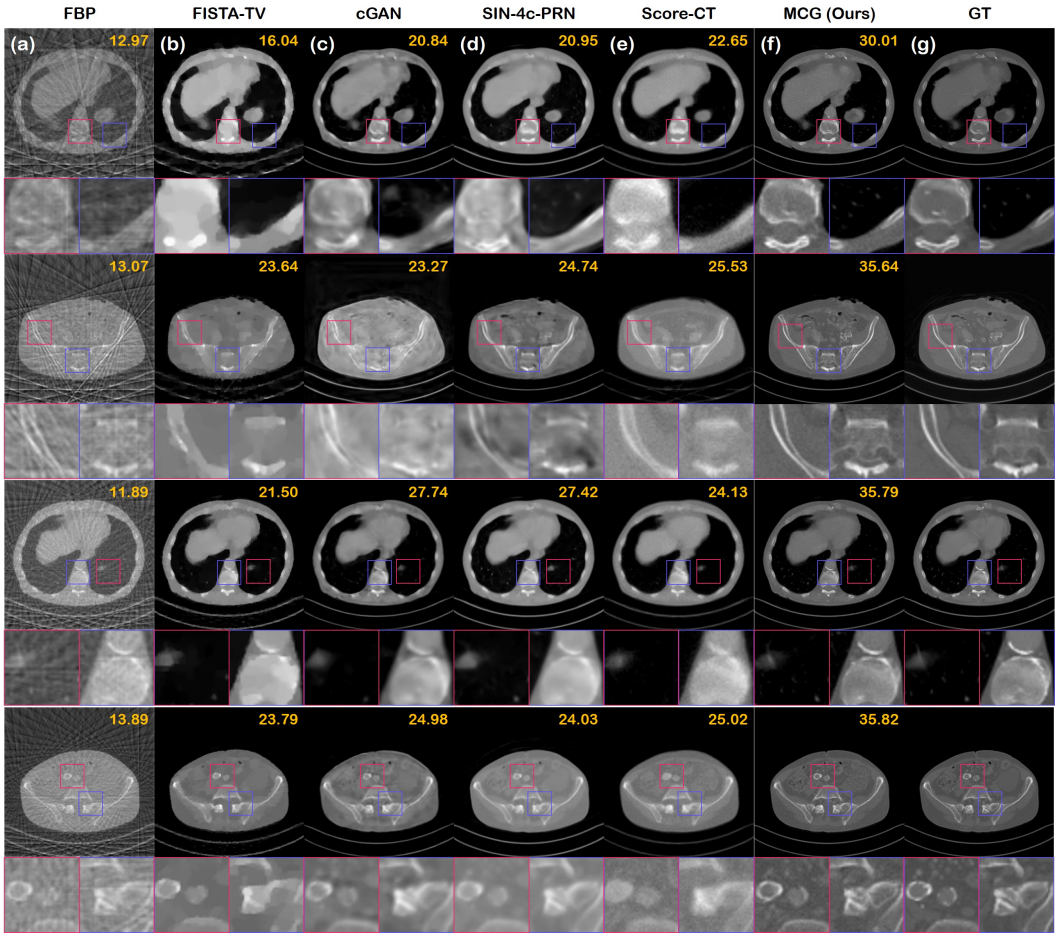


Figure 12: Sparse view CT reconstruction results on AAPM 256×256 data.(a) FBP, (b) FISTA-TV [3], (c) cGAN [15], (d) SIN-4c-PRN [50], (e) Score-CT [41], (f) MCG (Ours), (g) ground truth (GT).

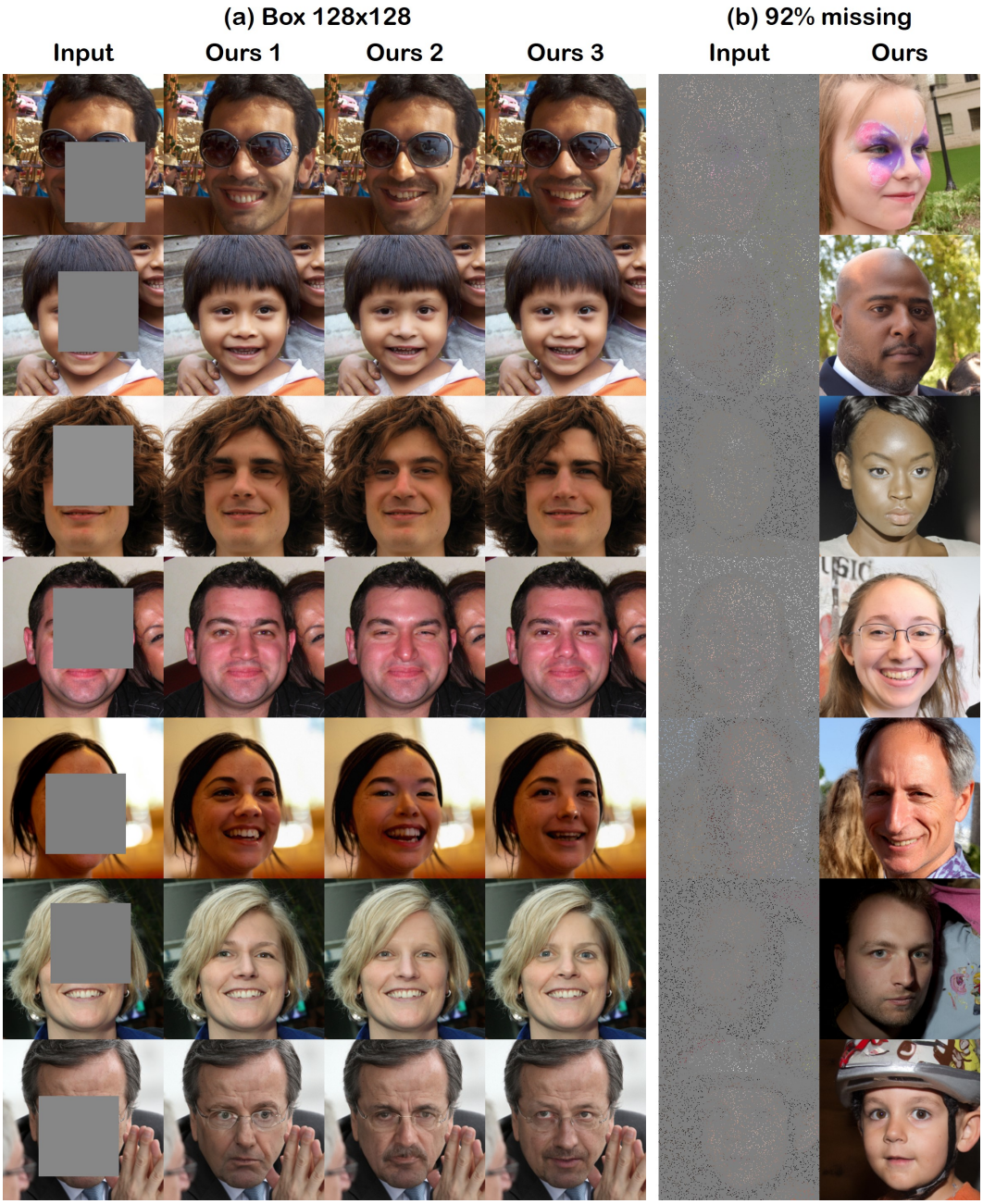


Figure 13: Inpainting results on FFHQ 256×256 data with MCG. (a) Inpainting of 128×128 box region. We show three stochastic samples generated with the proposed method. (b) 92 % pixel missing imputation.

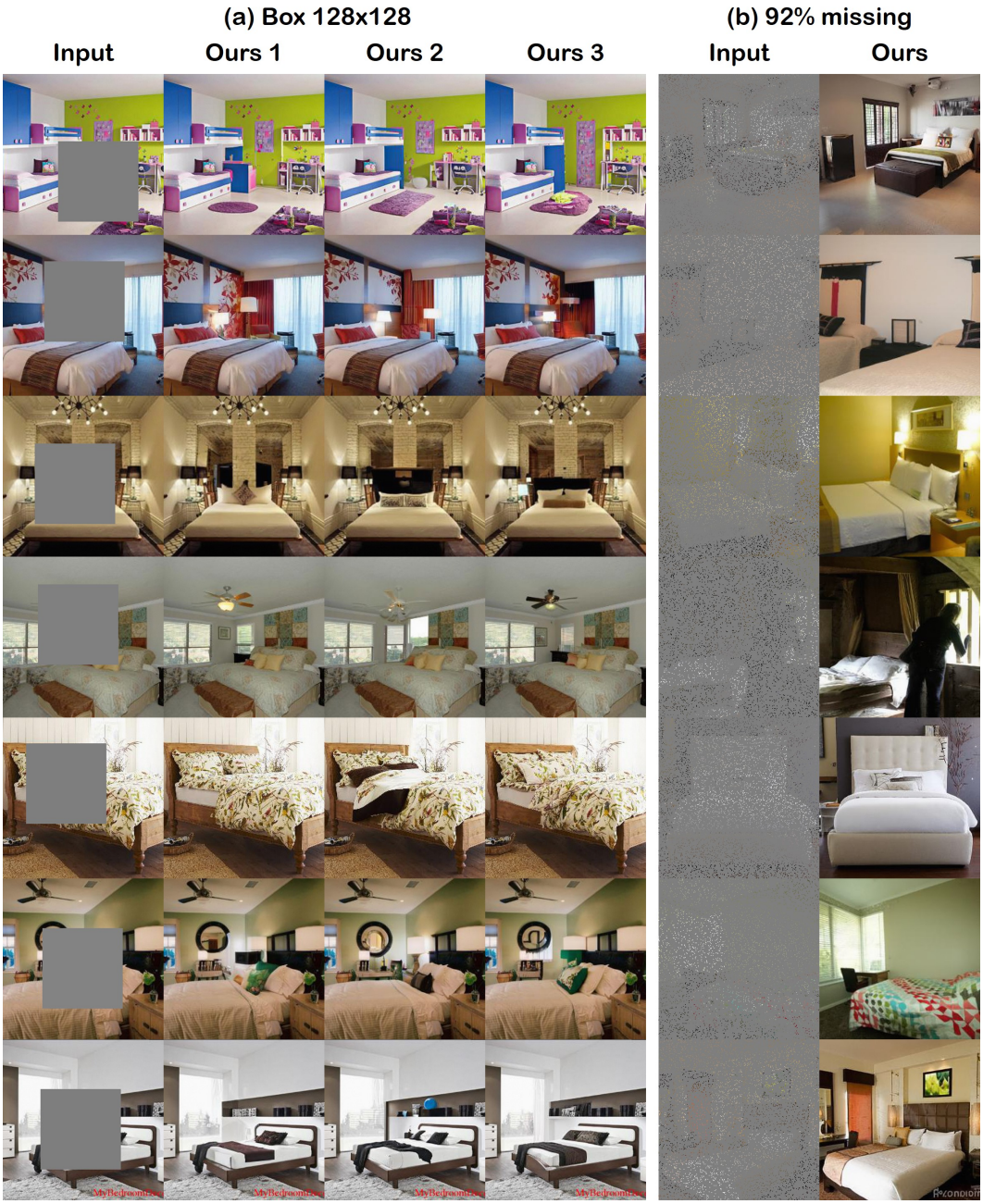


Figure 14: Inpainting results on LSUN-bedroom 256×256 data with MCG. (a) Inpainting of 128×128 box region. We show three stochastic samples generated with the proposed method. (b) 92 % pixel missing imputation.

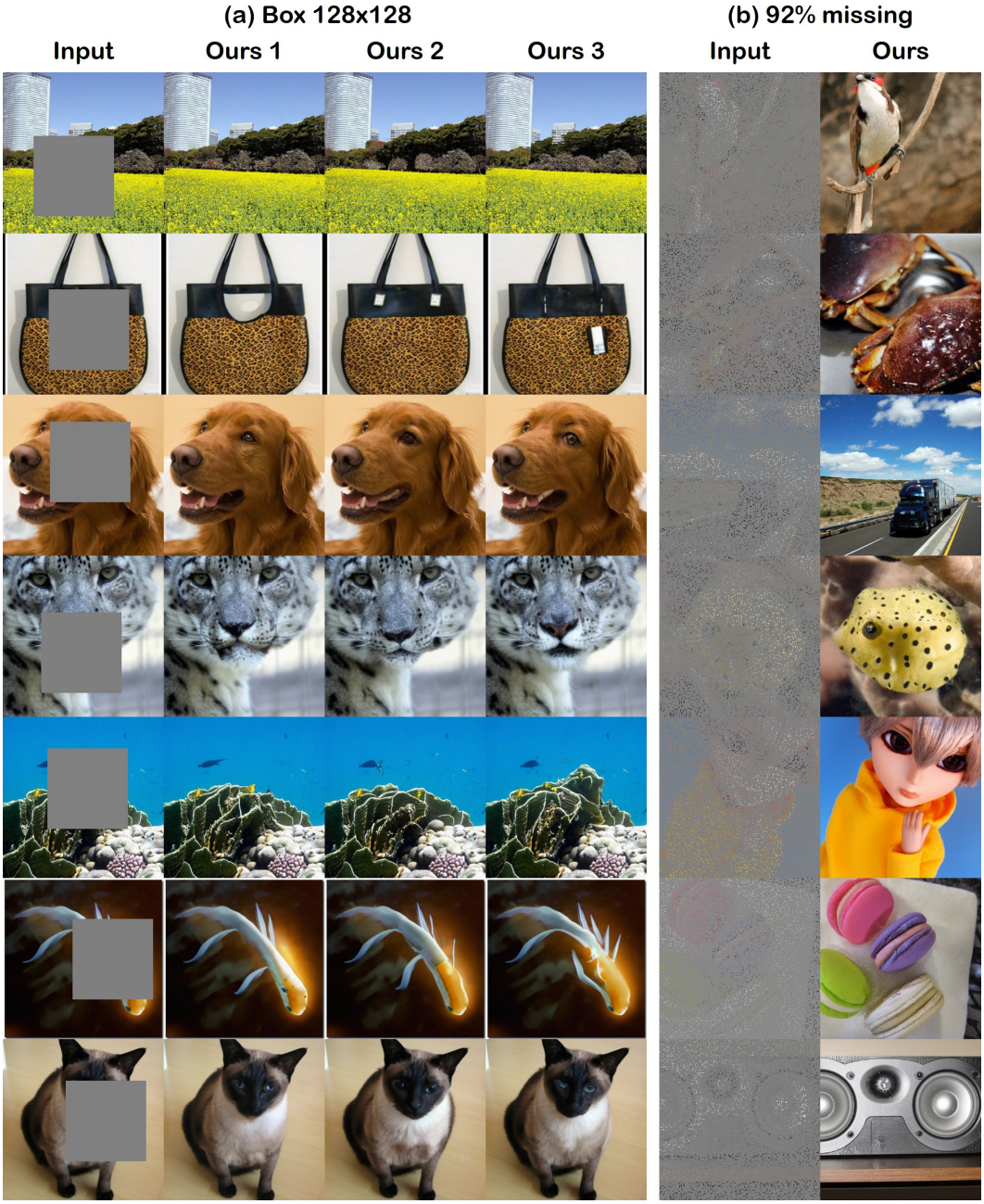


Figure 15: Inpainting results on ImageNet 256×256 data with MCG. (a) Inpainting of 128×128 box region. We show three stochastic samples generated with the proposed method. (b) 92 % pixel missing imputation.

Figure 16: Colorization results on (left) FFHQ 256×256 dataset, and (right) LSUN-bedroom 256×256 dataset. We show 3 different reconstructions for each measurement that are sampled with the proposed method.