GibbsDDRM: A Partially Collapsed Gibbs Sampler for Solving Blind Inverse Problems with Denoising Diffusion Restoration

Naoki Murata [1]   Koichi Saito [1]   Chieh-Hsin Lai [1]   Yuhta Takida [1]   Toshimitsu Uesaka [1]   Yuki Mitsufuji [1 2]
Stefano Ermon [3]

Abstract

Pre-trained diffusion models have been successfully used as priors in a variety of linear inverse problems, where the goal is to reconstruct a signal from noisy linear measurements. However, existing approaches require knowledge of the linear operator. In this paper, we propose GibbsDDRM, an extension of Denoising Diffusion Restoration Models (DDRM) to a blind setting in which the linear measurement operator is unknown. GibbsDDRM constructs a joint distribution of the data, measurements, and linear operator by using a pretrained diffusion model for the data prior, and it solves the problem by posterior sampling with an efficient variant of a Gibbs sampler. The proposed method is problem-agnostic, meaning that a pretrained diffusion model can be applied to various inverse problems without fine-tuning. In experiments, it achieved high performance on both blind image deblurring and vocal dereverberation tasks, despite the use of simple generic priors for the underlying linear operators.

1. Introduction

Inverse problems are frequently encountered in various science and engineering fields such as image processing, acoustic signal processing, and medical imaging. In an inverse problem, the goal is to restore a clean data signal from measurements generated by some forward (measurement) process. In image processing, problems such as deblurring (Zhu et al., 2018; Kupyn et al., 2019; Tu et al., 2022), inpainting (Yeh et al., 2017), and colorization (Larsson et al., 2016) are naturally formulated as inverse problems. In audio sig-
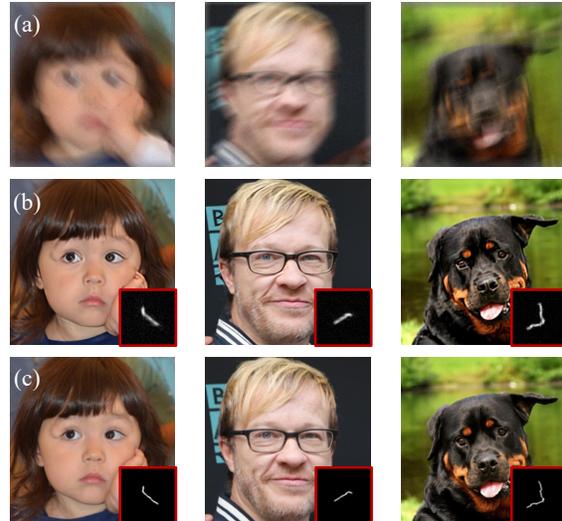

Figure 1. Blind image deblurring results obtained by GibbsDDRM: (a) measurement, (b) restored clean images with blur kernels (bottom right insets), and (c) ground truth images and blur kernels.

nal processing, problems such as dereverberation (Nakatani et al., 2010; Saito et al., 2023) and band extension (Larsen & Aarts, 2005) are also classic inverse problems. In medical imaging, many problems such as computed tomography (CT) (Zhu et al., 2018; Song et al., 2021a) also rely on inverse problem solving.

In general, inverse problems are ill-posed because the information in the original data is lost through the measurement process (e.g., because of noise); the incorporation of prior knowledge about the original data is thus critical. In the past, assumptions such as sparsity (Candès & Wakin, 2008), low rank (Fazel et al., 2008), and total variation (Candès et al., 2006) were made for the data distribution to narrow the set of plausible candidate solutions. A more recent trend has been to solve inverse problems by using richer deep generative models (Rick Chang et al., 2017; Anirudh et al., 2018; Kadkhodaie & Simoncelli, 2020; Whang et al., 2021) trained with a large amount of data as priors. In particular, the evolution of methods related to diffusion models (Kawar et al., 2021; 2022; Chung et al., 2023b;a) has been significant, and many such methods are problem-agnostic, meaning that

they do not require retraining of the generative model used for inference on each task (i.e., each inverse problem).

Existing approaches typically assume that the measurement process is known. However, many settings are blind, meaning that the measurement process itself is (partially) unknown. This is known as a blind setting and includes problems such as blind image deblurring (Pan et al., 2016) and audio dereverberation (Nakatani et al., 2010). For example, in a blind image deblurring problem, the original image has to be restored from the convolution process where the blur kernel is unknown. To address this additional uncertainty, priors are introduced on both the data and the parameters of the linear operator involved (Chan & Wong, 1998; Krishnan & Fergus, 2009; Xu et al., 2013). BlindDPS (Chung et al., 2023a) is a method that uses a pre-trained diffusion model for both data and parameters. However, while it can leverage widely available pre-trained diffusion models for signals such as images and audio, it requires training a diffusion model for the parameters of the linear operators of interest, severely restricting its applicability in practice.

To overcome this limitation, we propose GibbsDDRM, which does not require a data-driven prior model of the measurement process. This method is an extension of Denoising Diffusion Restoration Models (DDRM) (Kawar et al., 2022) – a method designed for non-blind linear inverse problems – to the blind linear setting. It constructs a joint distribution of the data, the measurements, and the linear operator's parameters by using a pre-trained diffusion model for the data and a generic prior for the measurement parameters. Then, it performs approximate sampling from the corresponding posterior distribution of the data and parameters conditioned on the measurements. Here, we adopt a partially collapsed Gibbs sampler (PCGS) (Van Dyk & Park, 2008) to enable efficient sampling from the posterior distribution. PCGS allows us to replace an intractable conditional distribution in the naïve Gibbs sampler with a more tractable one without changing the stationary distribution. PCGS alternately samples the data or latent variables and the linear operator's parameters, and the generative model's representational power is exploited while sampling the parameters of the linear operator. This allows our method to accurately estimate both data and the parameters despite using a simple prior for the parameters.

We conducted experiments on the tasks of blind image deblurring in the image processing domain and vocal dereverberation in the acoustic signal processing domain. The results confirm that high performance can be achieved on both tasks without strong assumptions on the prior for the linear operator's parameters. In the blind image deblurring task, GibbsDDRM demonstrates exceptional quantitative performance in terms of both image quality and faithfulness. It outperforms competing methods and BlindDPS by a large

margin in LPIPS, which measures the perceptual similarity of images. The results also show that a faithful image can be restored even with large measurement noise. (see Figure 1 for restored images and estimated blur kernels.) In vocal dereverberation, GibbsDDRM outperforms alternative methods in terms of the quality of the processed vocal, the proximity of the signals, and the degree of reverberation removal.

2. Background

Blind linear inverse problems. Blind linear inverse problems involve the estimation of both unknown clean data and the parameters of a linear operator from noisy measurements. This type of problem can be formulated as a linear system of equations of the following form:

$$\mathbf{y} = \mathbf{H}_{\boldsymbol{\varphi}}\mathbf{x}_0 + \mathbf{z}, \qquad (1)$$

where $\mathbf{y} \in \mathbb{R}^{d_\mathbf{y}}$ is a vector of measurements, $\mathbf{H}_{\boldsymbol{\varphi}} \in \mathbb{R}^{d_\mathbf{y} \times d_{\mathbf{x}_0}}$ is a linear operator parameterized by $\boldsymbol{\varphi} \in \mathbb{R}^{d_\varphi}$, and $\mathbf{x}_0 \in \mathbb{R}^{d_{\mathbf{x}_0}}$ is the unknown original clean data to be estimated. $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma_\mathbf{y}^2\mathbf{I})$ is a Gaussian measurement noise with known covariance $\sigma_\mathbf{y}^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. For notational convenience, we index the clean data $\mathbf{x}_0$ with "0" to distinguish it from latent variables of the diffusion model that are defined later. The aim here is to find estimates of both $\mathbf{x}_0$ and $\boldsymbol{\varphi}$ that fit the given noisy measurements $\mathbf{y}$. The problem is ill-posed without any additional assumptions. To obtain a solution, it is assumed that $\mathbf{x}_0$ is drawn from a generative model $p_\theta(\mathbf{x}_0)$ (close to the true data distribution), and that the parameters $\boldsymbol{\varphi}$ are drawn from a known prior $p(\boldsymbol{\varphi})$ independently of the data. In the Bayesian framework, the optimal solution is to sample from the posterior $p(\mathbf{x}_0, \boldsymbol{\varphi}|\mathbf{y})$.

Denoising Diffusion Probabilistic Models. Denoising Diffusion Probabilistic Models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019; Song et al., 2021b; Lai et al., 2022), or diffusion models for short, are generative models with a Markov chain $\mathbf{x}_T \to \cdots \to \mathbf{x}_t \to \cdots \to \mathbf{x}_0$ represented by the following joint distribution:

$$p_\theta(\mathbf{x}_{0:T}) = p_\theta^{(T)}(\mathbf{x}_T) \prod_{t=0}^{T-1} p_\theta^{(t)}(\mathbf{x}_t|\mathbf{x}_{t+1}), \qquad (2)$$

where the model's output is $\mathbf{x}_0$. To train a diffusion model, a fixed variational inference distribution is introduced:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = q^{(T)}(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=1}^{T-1} q^{(t)}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0), \quad (3)$$

which gives the evidence lower bound (ELBO) on the maximum likelihood objective. With Gaussian parameterization

for p_θ and q, the ELBO objective is reduced to the following denoising autoencoder objective:

$$\sum_{t=1}^{T} \gamma_t \mathbb{E}_{(\mathbf{x}_0, \mathbf{x}_t) \sim p_{\text{data}}(\mathbf{x}_0) q(\mathbf{x}_t | \mathbf{x}_0)} \left[\| \mathbf{x}_0 - f_\theta^{(t)}(\mathbf{x}_t) \|_2^2 \right]. \quad (4)$$

Here, $f_\theta^{(t)}$ is a θ-parameterized neural network that estimates noiseless data $\mathbf{x}_0$ from noisy $\mathbf{x}_t$ and characterizes p_θ; $\mathbf{x}_{\theta,t}$ denotes the estimate of noise-less data by $f_\theta^{(t)}$; γ_t are positive weighting coefficients determined by q.

Denoising Diffusion Restoration Models. Denoising Diffusion Restoration Models (DDRM) (Kawar et al., 2022) is a method that uses a pre-trained diffusion model as a prior for data in a non-blind linear inverse problem. It is defined as a Markov chain $\mathbf{x}_T \to \mathbf{x}_{T-1} \to \cdots \to \mathbf{x}_1 \to \mathbf{x}_0$ (where $\mathbf{x}_t \in \mathbb{R}^{d_{\mathbf{x}_0}}$) conditioned on the measurements $\mathbf{y}$:

$$p(\mathbf{x}_{0:T} | \mathbf{y}) = p_\theta^{(T)}(\mathbf{x}_T | \mathbf{y}) \prod_{t=0}^{T-1} p_\theta^{(t)}(\mathbf{x}_t | \mathbf{x}_{t+1}, \mathbf{y}), \quad (5)$$

where $\mathbf{x}_0$ is the model's output. The conditionals in DDRM are defined in terms of the denoising function $f_\theta^{(t)}$ of a pre-trained diffusion model; intriguingly, the objective derived using the ELBO coincides with that of the unconditional diffusion model, except for a constant factor. This means that the unconditionally pre-trained diffusion model can be used during inference without finetuning. The core idea of DDRM is to use the singular value decomposition (SVD) of a linear operator $\mathbf{H}$ to transform both the unknown input $\mathbf{x}_0$ and the observed output $\mathbf{y}$, potentially corrupted by noise, to a shared spectral space. In this space, DDRM executes denoising on dimensions for which information from $\mathbf{y}$ is available (i.e., when the singular values are non-zero). When such information is not available (i.e., when the singular values are zero or the noise in the dimension is large), DDRM performs imputation while explicitly considering the measurement noise.

Partially collapsed Gibbs sampler. A Gibbs sampler is a simple, widely used Markov chain Monte Carlo method for sampling from the joint distribution of a set of variables (Casella & George, 1992). The procedure entails iterative sampling from the fully conditional distributions of each variable, given the current values of the other variables. A blocked Gibbs sampler (Liu et al., 1994) is a variant in which, instead of sampling each variable individually, variables in a group or a "block" of variables are sampled simultaneously while conditioned on all the other variables. This approach is effective when the variables within a block are highly correlated, and it can improve the sampler's convergence speed.

A partially collapsed Gibbs sampler (PCGS) (Van Dyk & Park, 2008; Kail et al., 2012) is a generalization of a blocked
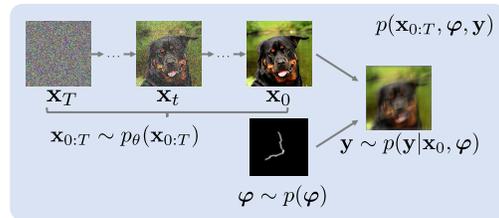

Figure 2. Graphical model for the joint distribution in Eq. (7).

Gibbs sampler that effectively explores the probability space through three basic operations in the sampling procedure: *marginalization*, *permutation*, and *trimming*, which are described in detail in (Van Dyk & Park, 2008) and Appendix A. In short, the removal of certain variables among the conditional variables does not alter the Gibbs sampler's stationary distribution, as long as these variables are not included among the conditional variables until the next time they are sampled. Hence, we can achieve efficient sampling when the distributions obtained after trimming are tractable.

3. GibbsDDRM: Partially Collapsed Gibbs Sampler with DDRM

3.1. Target joint distribution for blind linear inverse problems

In this paper, we seek to solve blind linear inverse problems by sampling from the posterior of the joint distribution of the data and the linear operator's parameters, given the measurements. The joint distribution of the data $\mathbf{x}_0$, parameters $\boldsymbol{\varphi}$, and measurements $\mathbf{y}$ is defined as follows:

$$p(\mathbf{x}_0, \mathbf{y}, \boldsymbol{\varphi}) = p_\theta(\mathbf{x}_0) p(\boldsymbol{\varphi}) \mathcal{N}(\mathbf{y} | \mathbf{H}_{\boldsymbol{\varphi}} \mathbf{x}_0, \sigma_{\mathbf{y}}^2 \mathbf{I}), \quad (6)$$

where $p_\theta(\mathbf{x}_0)$ and $p(\boldsymbol{\varphi})$ are the known prior distributions for the data and the parameters, respectively. The Gaussian distribution $\mathcal{N}(\mathbf{y} | \mathbf{H}_{\boldsymbol{\varphi}} \mathbf{x}_0, \sigma_{\mathbf{y}}^2 \mathbf{I})$ comes from the measurement model given in Eq. (1). The aim is to sample from the joint posterior distribution $p(\mathbf{x}_0, \boldsymbol{\varphi} | \mathbf{y})$. Using a pre-trained generative model as a prior $p_\theta(\mathbf{x}_0)$ can drastically improve the solutions in inverse problems; however, inference can be challenging. Even in the non-blind setting where $\boldsymbol{\varphi}$ is known, sampling from the posterior is intractable and requires approximations like in DDRM (Kawar et al., 2022).

Here we model the data distribution using a pre-trained diffusion model as in Eq. (2). This leads to the following joint distribution over the data, its latent variables, and the parameters, as shown in Figure 2,

$$p(\mathbf{x}_{0:T}, \boldsymbol{\varphi}, \mathbf{y})$$
$$= p_\theta^{(T)}(\mathbf{x}_T) \prod_{t=0}^{T-1} p_\theta^{(t)}(\mathbf{x}_t | \mathbf{x}_{t+1}) p(\boldsymbol{\varphi}) \mathcal{N}(\mathbf{y} | \mathbf{H}_{\boldsymbol{\varphi}} \mathbf{x}_0, \sigma_{\mathbf{y}}^2 \mathbf{I}).$$
$$(7)$$

Note that sampling from the posterior distribution $p(\mathbf{x}_{0:T}|\boldsymbol{\varphi}, \mathbf{y})$ under a fixed $\boldsymbol{\varphi}$ corresponds to the objective of DDRM. In addition, we also assume that the parameters' prior $p(\boldsymbol{\varphi})$ is a generic and simple prior, such as a sparsity prior.

3.2. Partially Collapsed Gibbs Sampler for the joint distribution

To sample from the joint posterior in Eq. (7), we could attempt to sample from the joint posterior distribution that includes the latent variables of the diffusion model. However, it is still not feasible to run a naïve Gibbs sampler for the posterior $p(\mathbf{x}_{0:T}, \boldsymbol{\varphi}|\mathbf{y})$, as it would require a conditional distribution for every individual variable, conditioned on all the other variables. For instance, the conditional distribution $p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{x}_{t+1:T}, \boldsymbol{\varphi}, \mathbf{y})$ for the joint distribution defined in Eq. (7) is not obvious.

A possible strategy is to use a blocked Gibbs sampler (Liu et al., 1994) with the variables divided into two groups, $\mathbf{x}_{0:T}$ and $\boldsymbol{\varphi}$, and sampled alternately. In more detail, after initializing $\boldsymbol{\varphi}$, the sampling procedure of DDRM is performed keeping $\boldsymbol{\varphi}$ fixed to obtain an estimate of the clean data $\mathbf{x}_0$. Then, $\boldsymbol{\varphi}$ is sampled such that it is consistent with the estimated data $\mathbf{x}_0$ and measurements $\mathbf{y}$. By repeating these operations, we can sample $\mathbf{x}_0$ and $\boldsymbol{\varphi}$ from the joint posterior. However, this approach may be inefficient because of the small number of updates made to $\boldsymbol{\varphi}$: the entire sampling of $\mathbf{x}_{0:T}$ must be performed for a step of sampling $\boldsymbol{\varphi}$, which results in slow convergence.

Hence, we adopt a partially collapsed Gibbs sampler (PCGS) (Van Dyk & Park, 2008) for the joint posterior. This strategy's main advantage is that we can still use a similar sampling method defined by the original DDRM. This enables simultaneous sampling of the latent variables $\mathbf{x}_{1:T}$ and the linear operator's parameters $\boldsymbol{\varphi}$ within a cycle of DDRM sampling, thus improving the convergence speed.

In a naïve Gibbs sampler, the order of sampling variables is arbitrary. In a PCGS, however, the sampling order must be carefully chosen to facilitate the trimming operation, which removes conditional variables from the conditional distribution. Specifically, once a variable has been marginalized and removed from the conditional set, it should not be added back until the next time it is sampled. We show a simple example of a PCGS in Appendix A. Figure 3 shows the sampling order of the proposed PCGS. After sampling $\mathbf{x}_T$, the following operations are performed in descending order of t, until $t = 0$: for each t, $\mathbf{x}_t$ is sampled once, and then $\boldsymbol{\varphi}$ and $\mathbf{x}_t$ are alternately sampled M_t times. One set of these operations constitutes a single cycle of the PCGS, and the operations are repeated for N cycles.
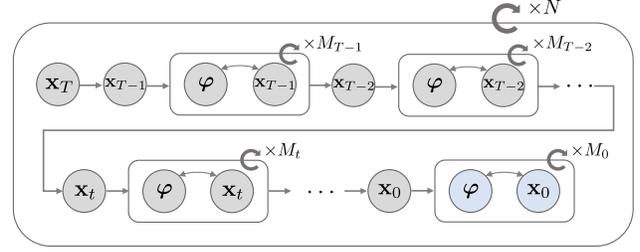
The proposed PCGS is defined in Algorithm 1. The fol-

Figure 3. Sampling order of variables in the proposed PCGS, whose output entails the final sample of data $\mathbf{x}_0$ and parameters $\boldsymbol{\varphi}$.

lowing proposition ensures that it samples from the true posterior distribution.

Proposition 3.1. *The PCGS defined in Algorithm 1 has the true posterior distribution $p(\mathbf{x}_{0:T}, \boldsymbol{\varphi}|\mathbf{y})$ as its stationary distribution if the approximations to the conditional distributions are exact.*

We give the proof in Appendix A.

Algorithm 1 Proposed PCGS for the posterior in Eq. (7)

Input: Measurement $\mathbf{y}$, initial values $\boldsymbol{\varphi}^{(0,0)}$.
Output: Restored data $\mathbf{x}_0^{(N,M_0)}$, linear operator's parameters $\boldsymbol{\varphi}^{(N,K)}$.
$K \leftarrow 0$ // K counts the number of updates for $\boldsymbol{\varphi}$ in a cycle.
for $n = 1$ **to** N **do**
  $\boldsymbol{\varphi}^{(n,0)} \leftarrow \boldsymbol{\varphi}^{(n-1,K)}, K \leftarrow 0$
  Sample $\mathbf{x}_T^{(n,0)} \sim p(\mathbf{x}_T|\boldsymbol{\varphi}^{(n,K)}, \mathbf{y})$
  // $\uparrow$ approximated by $p_\theta(\mathbf{x}_T|\boldsymbol{\varphi}, \mathbf{y})$.
  for $t = T - 1$ **to** 0 **do**
    $\chi_t \leftarrow \{\mathbf{x}_{t+1}^{(n,M_{t+1})}, \mathbf{x}_{t+2}^{(n,M_{t+2})}, \cdots, \mathbf{x}_T^{(n,0)}\}$
    Sample $\mathbf{x}_t^{(n,0)} \sim p(\mathbf{x}_t|\boldsymbol{\varphi}^{(n,K)}, \chi_t, \mathbf{y})$
    // $\uparrow$ approximated by $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}, \boldsymbol{\varphi}, \mathbf{y})$.
    for $m = 1$ **to** M_t **do**
      Sample $\boldsymbol{\varphi}^{(n,K+1)} \sim p(\boldsymbol{\varphi}|\mathbf{x}_t^{(n,m-1)}, \chi_t, \mathbf{y})$
      // $\uparrow$ Langevin sampling with the approximated score $\nabla_{\boldsymbol{\varphi}} \log p(\mathbf{y}|\mathbf{x}_{\theta,t}, \boldsymbol{\varphi})$.
      $K \leftarrow K + 1$
      Sample $\mathbf{x}_t^{(n,m)} \sim p(\mathbf{x}_t|\boldsymbol{\varphi}^{(n,K)}, \chi_t, \mathbf{y})$
      // $\uparrow$ approximated by $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}, \boldsymbol{\varphi}, \mathbf{y})$.
    end for
  end for
end for

Proposition 3.1 states that it is possible to sample reasonable data and parameters by executing the PCGS defined in Algorithm 1, but the conditional distributions the PCGS includes are intractable. Hence, we replace each conditional distribution with approximations from which we can efficiently sample. In the following paragraphs, we provide the details of the sampling procedures at each step.

Sampling of $\mathbf{x}_T$. The sampling of $\mathbf{x}_T$ is performed with the distribution $p(\mathbf{x}_T | \boldsymbol{\varphi}, \mathbf{y})$, which is obtained by trimming $\mathbf{x}_{0:T-1}$. Because this conditional distribution is intractable, as discussed above, we use modified DDRM to approximate the conditional distribution.

Here, in order to introduce the modified DDRM, we use SVD of the linear operator $\mathbf{H}_{\boldsymbol{\varphi}}$ and its spectral space, similarly to previous studies (Kawar et al., 2021; 2022). The SVD is given as $\mathbf{H}_{\boldsymbol{\varphi}} = \mathbf{U}_{\boldsymbol{\varphi}} \boldsymbol{\Sigma}_{\boldsymbol{\varphi}} \mathbf{V}_{\boldsymbol{\varphi}}^{\mathsf{T}}$, where $\mathbf{U}_{\boldsymbol{\varphi}} \in \mathbb{R}^{d_{\mathbf{y}} \times d_{\mathbf{y}}}$ and $\mathbf{V}_{\boldsymbol{\varphi}} \in \mathbb{R}^{d_{\mathbf{x}_0} \times d_{\mathbf{x}_0}}$ are orthogonal matrices, and $\boldsymbol{\Sigma}_{\boldsymbol{\varphi}} \in \mathbb{R}^{d_{\mathbf{y}} \times d_{\mathbf{x}_0}}$ is a rectangular diagonal matrix. Here we assume $d_{\mathbf{y}} \leq d_{\mathbf{x}_0}$, but our method would work for $d_{\mathbf{y}} > d_{\mathbf{x}_0}$. The diagonal elements of $\boldsymbol{\Sigma}_{\boldsymbol{\varphi}}$ are the singular values of $\mathbf{H}_{\boldsymbol{\varphi}}$ in descending order, denoted $s_{1,\boldsymbol{\varphi}}, s_{2,\boldsymbol{\varphi}}, \cdots, s_{d_{\mathbf{y}},\boldsymbol{\varphi}}$. Hereafter, we omit the subscript $\boldsymbol{\varphi}$ from the singular values for notational simplicity. The values in the spectral space are represented as follows: $\overline{\mathbf{x}}_t^{(i)}$ is the i-th element of $\overline{\mathbf{x}}_t = \mathbf{V}_{\boldsymbol{\varphi}}^{\mathsf{T}} \mathbf{x}_t$, and $\overline{\mathbf{y}}^{(i)}$ is the i-th element of $\overline{\mathbf{y}} = \boldsymbol{\Sigma}_{\boldsymbol{\varphi}}^{\dagger} \mathbf{U}_{\boldsymbol{\varphi}}^{\mathsf{T}} \mathbf{y}$, where $\mathbf{A}^{\dagger}$ is the Moore-Penrose pseudo-inverse of a matrix $\mathbf{A}$. Note that the spectral space also depends on the parameters $\boldsymbol{\varphi}$, which is unknown in our blind setting, unlike in DDRM. Our modified DDRM update for sampling $\mathbf{x}_T$ is defined as follows:

$$p_{\theta}^{(T)} \left(\overline{\mathbf{x}}_T^{(i)} \mid \mathbf{y}, \boldsymbol{\varphi} \right) =$$
$$\begin{cases} \mathcal{N} \left(\overline{\mathbf{y}}^{(i)}, \sigma_T^2 - \sigma_{\mathbf{y}}^2 / s_i^2 \right) & \text{if } s_i > 0 \\ \mathcal{N} \left(0, \sigma_T^2 \right) & \text{if } s_i = 0 \end{cases}, \quad (8)$$

where the only difference from the original DDRM is that the parameters $\boldsymbol{\varphi}$ are treated as random variables.

Sampling of $\mathbf{x}_t$. The sampling of $\mathbf{x}_t$ ($t < T$) is performed by sampling from the conditional distribution $p(\mathbf{x}_t | \mathbf{x}_{t+1:T}, \boldsymbol{\varphi}, \mathbf{y})$, which trims $\mathbf{x}_{0:t-1}$ if $t > 0$. As in the sampling of $\mathbf{x}_T$, we approximate the conditional distribution by modifying DDRM. Denoting the prediction of $\mathbf{x}_0$ at every time step t by $\mathbf{x}_{\theta,t}$ which is made by the diffusion model as in Sec. 2, modified DDRM is defined as follows:

$$p_{\theta}^{(t)} \left(\overline{\mathbf{x}}_t^{(i)} \mid \mathbf{x}_{t+1}, \boldsymbol{\varphi}, \mathbf{y} \right) =$$
$$\begin{cases} \mathcal{N} \left(\overline{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1 - \eta^2} \sigma_t \frac{\overline{\mathbf{x}}_{t+1}^{(i)} - \overline{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{t+1}}, \eta^2 \sigma_t^2 \right) & \text{if } s_i = 0 \\ \mathcal{N} \left(\overline{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1 - \eta^2} \sigma_t \frac{\overline{\mathbf{y}}^{(i)} - \overline{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{\mathbf{y}}/s_i}, \eta^2 \sigma_t^2 \right) & \text{if } \sigma_t < \frac{\sigma_{\mathbf{y}}}{s_i} \\ \mathcal{N} \left((1 - \eta_b) \overline{\mathbf{x}}_{\theta,t}^{(i)} + \eta_b \overline{\mathbf{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_{\mathbf{y}}^2}{s_i^2} \eta_b^2 \right) & \text{if } \sigma_t \geq \frac{\sigma_{\mathbf{y}}}{s_i} \end{cases}$$
$$(9)$$

where $0 \leq \eta \leq 1$ and $0 \leq \eta_b \leq 1$ are hyperparameters, and $0 = \sigma_0 < \sigma_1 < \sigma_2 < \cdots < \sigma_T$ are noise levels that is the same as that defined with the pre-trained diffusion model.

Thus we have the approximation

$$p(\mathbf{x}_t | \mathbf{x}_{t+1:T}, \boldsymbol{\varphi}, \mathbf{y}) \simeq p_{\theta}(\mathbf{x}_t | \mathbf{x}_{t+1:T}, \boldsymbol{\varphi}, \mathbf{y})$$
$$= p_{\theta}(\mathbf{x}_t | \mathbf{x}_{t+1}, \boldsymbol{\varphi}, \mathbf{y}), \quad (10)$$

where the final equation comes from the Markov property of the modified DDRM.

Sampling of $\boldsymbol{\varphi}$. At time step t, the sampling of the parameters $\boldsymbol{\varphi}$ is done by using the conditional distribution $p(\boldsymbol{\varphi} | \mathbf{x}_{t:T}, \mathbf{y})$. For the joint distribution defined by Eq. (7), the conditional distribution is not easily obtained because, while $\boldsymbol{\varphi}$ and $\mathbf{x}_{t:T}$ are related through $\mathbf{x}_0$, the distribution of $\mathbf{x}_0$ cannot be evaluated at this point. Hence, we use the approximation in (Chung et al., 2023b;a) for the score of the conditional distribution and then perform sampling by Langevin dynamics (Langevin, 1908), as follows:

$$\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} + (\xi/2) \nabla_{\boldsymbol{\varphi}} \log p(\boldsymbol{\varphi} | \mathbf{x}_{t:T}, \mathbf{y}) + \sqrt{\xi} \boldsymbol{\epsilon}, \quad (11)$$

where ξ is a step size and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. By Bayes' rule, the score $\nabla_{\boldsymbol{\varphi}} \log q(\boldsymbol{\varphi} | \mathbf{x}_{t:T}, \mathbf{y})$ can be decomposed into two terms:

$$\nabla_{\boldsymbol{\varphi}} \log p(\boldsymbol{\varphi} | \mathbf{x}_{t:T}, \mathbf{y}) =$$
$$\nabla_{\boldsymbol{\varphi}} \log p(\mathbf{y} | \mathbf{x}_{t:T}, \boldsymbol{\varphi}) + \nabla_{\boldsymbol{\varphi}} \log p(\boldsymbol{\varphi} | \mathbf{x}_{t:T}). \quad (12)$$

Regarding the first term, we exploit the following theorem.

Theorem 3.2. *(modified version of Theorem 1 in (Chung et al., 2023b)) For the measurement model in Eq. (1), we have*

$$p(\mathbf{y} | \mathbf{x}_{t:T}, \boldsymbol{\varphi}) \simeq p(\mathbf{y} | \mathbf{x}_{\theta,t}, \boldsymbol{\varphi}), \quad (13)$$

and the approximation error can be quantified with the Jensen gap (Gao et al., 2017), which is upper bounded by

$$\mathcal{J} \leq \frac{1}{\sigma_{\mathbf{y}} \left(\sqrt{2\pi\sigma_{\mathbf{y}}^2} \right)^{d_{\mathbf{y}}}} e^{-1/2} s_1 m_1, \quad (14)$$

where $m_1 := \int \|\mathbf{x}_0 - \mathbf{x}_{\theta,t}\| p(\mathbf{x}_0 | \mathbf{x}_{t:T}) d\mathbf{x}_0$, and s_1 is the largest singular value of $\mathbf{H}_{\boldsymbol{\varphi}}$.

By leveraging Theorem 3.2, we obtain the approximate gradient with respect to $\boldsymbol{\varphi}$ for the Langevin dynamics:

$$\nabla_{\boldsymbol{\varphi}} \log p(\mathbf{y} | \mathbf{x}_{t:T}, \boldsymbol{\varphi}) \simeq \nabla_{\boldsymbol{\varphi}} \log p(\mathbf{y} | \mathbf{x}_{\theta,t}, \boldsymbol{\varphi}), \quad (15)$$

and for our measurement model in Eq. (1), the gradient is

$$\nabla_{\boldsymbol{\varphi}} \log p(\mathbf{y} | \mathbf{x}_{\theta,t}, \boldsymbol{\varphi}) = -\frac{1}{2\sigma_{\mathbf{y}}^2} \nabla_{\boldsymbol{\varphi}} \|\mathbf{y} - \mathbf{H}_{\boldsymbol{\varphi}} \mathbf{x}_{\theta,t}\|_2^2, \quad (16)$$

which is tractable in practice.

As for the second term in Eq. (12), the conditional variables can be eliminated since $\mathbf{x}_{t:T}$ and $\boldsymbol{\varphi}$ are independent from
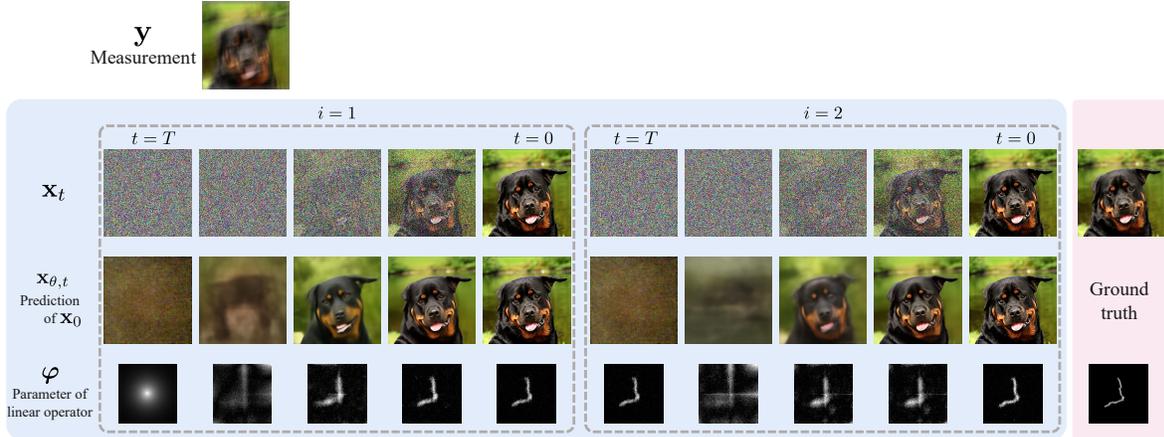
Figure 4. Visualization of GibbsDDRM for the blind image deblurring task on the AFHQ dataset.

Eq. (7). As a result, we can use a simple prior distribution (e.g., a Gaussian prior) for $\boldsymbol{\varphi}$ that does not depend on $\mathbf{x}_{t:T}$.

We now have the conditional score of $\boldsymbol{\varphi}$ for the Langevin dynamics as follows:

$$
\begin{aligned}
&\nabla_{\boldsymbol{\varphi}} \log p(\boldsymbol{\varphi}|\mathbf{x}_{t:T}, \mathbf{y}) \\
&\simeq -\frac{1}{2\sigma_{\mathbf{y}}^2} \nabla_{\boldsymbol{\varphi}} \|\mathbf{y} - \mathbf{H}_{\boldsymbol{\varphi}} \mathbf{x}_{\theta,t}\|_2^2 + \nabla_{\boldsymbol{\varphi}} \log p(\boldsymbol{\varphi}).
\end{aligned}
\quad (17)
$$

Note that at a particular time step t, $\mathbf{x}_t$ varies because of the Gibbs sampling, and so does $\mathbf{x}_{\theta,t}$. This iterative process can be viewed as feeding the information from the diffusion model to the parameter estimation. It allows for accurate parameter estimation even with simple priors.

We refer to the proposed PCGS as the Gibbs Denoising Diffusion Restoration Models (GibbsDDRM), and we describe the details of its instantiation for each of our experimental tasks in Appendix B.

3.3. Implementation considerations

Initialization of $\boldsymbol{\varphi}$. In GibbsDDRM, the initialization for $\boldsymbol{\varphi}$ is arbitrary. If an existing simple method can be used to obtain an estimate of $\boldsymbol{\varphi}$, then we can use that estimate as the initial value. In our experiments, we initialize the blur kernel with a Gaussian blur kernel in the blind image deblurring task. For the vocal dereverberation task, the parameters are initialized with estimates obtained by the weighted prediction error method (WPE) (Nakatani et al., 2010), which is an unsupervised method that is not based on machine learning, to accelerate the convergence speed.

Dependence of number of iterations, M_t, on time step. When t is large, the estimation of $\mathbf{x}_0$ ($= \mathbf{x}_{\theta,t}$) is difficult because of the large amount of noise in $\mathbf{x}_t$. This uncertainty can lead to instability in the sampling of $\boldsymbol{\varphi}$. The number

of sampling steps for $\boldsymbol{\varphi}$ can vary across the diffusion time steps and may even be zero. Accordingly, we use a strategy of not updating $\boldsymbol{\varphi}$ when t is large.

4. Experiments

We demonstrate our approach through two tasks: blind image deblurring in the image processing domain and vocal dereverberation in the audio processing domain.

4.1. Blind image deblurring.

The aim of blind image deblurring is to restore a clean image from a noisy blurred image without knowledge of the blur kernel. The details of the problem formulation and its instantiation as a linear inverse problem are given in Appendix B. Our code is available at `https://github.com/sony/gibbsddrm`.

Experimental settings. We conduct experiments on the Flickr Face High Quality (FFHQ) 256×256 dataset (Karras et al., 2019) and the Animal Faces-HQ (AFHQ) 256×256 dataset (Choi et al., 2020b). We use a 1000-image validation set for FFHQ and a 500-image test set for the dog class in AFHQ. All images are normalized to the range $[0, 1]$. The blur type used is motion blur, and blur kernels of size 64×64 are generated via code [1], with an intensity value of 0.5. We use the pre-trained diffusion models from (Choi et al., 2021) [2] for FFHQ and from (Dhariwal & Nichol, 2021) for AFHQ, without finetuning for this task. Measurements are generated by convolving the blur kernel with a ground truth image and adding Gaussian noise with $\sigma_{\mathbf{y}} = 0.02$. We use $\eta = 0.80$ and $\eta_b = 0.90$ for the pro-

---

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

[2] `https://github.com/jychoi118/ilvr_adm`

Table 1. Blind image deblurring results on FFHQ and AFHQ (256×256). The blurred images have additive Gaussian noise with $\sigma_{\mathbf{y}} = 0.02$. (∗) The results for BlindDPS (Chung et al., 2023a), as reported in the original paper, are also listed, although that method uses a pre-trained score function for blur kernels. The results of DDRM (Kawar et al., 2022) with the ground truth kernels (i.e., non-blind setting) are also listed. **Bold**: Best. underscore: second best.

Method	**FFHQ** (256×256)			**AFHQ** (256×256)		
	FID↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS↓	PSNR↑
GibbsDDRM (ours)	38.71	**0.115**	25.80	48.00	**0.197**	22.01
MPRNet (Zamir et al., 2021)	62.92	0.211	**27.23**	50.43	0.278	**27.02**
DeblurGANv2 (Kupyn et al., 2019)	141.55	0.320	19.86	156.92	0.429	17.64
Pan-DCP (Pan et al., 2017)	239.69	0.653	14.20	185.40	0.632	14.48
SelfDeblur (Ren et al., 2020)	283.69	0.859	10.44	250.20	0.840	10.34
BlindDPS (Chung et al., 2023a)∗	**29.49**	0.281	22.24	**23.89**	0.338	20.92
DDRM (Kawar et al., 2022) with GT kernel	33.97	0.062	30.64	24.60	0.078	29.37

posed method. The number of steps, T, is set to 100, and N is set to 1. Following the discussion in Section 3.3, M_t is set to 0 for $70 \le t \le 100$ and to 3 for $t < 70$. The number of iterations and the step size for Langevin dynamics (Eq. (16)) are set to 500 and 1.0×10^{-11}, respectively. The blur kernel is initialized with a Gaussian blur kernel and normalized to have non-negative values and a sum of 1.0, which remains normalized throughout the processing. We use a Laplace prior for the parameters φ, which has the form $\nabla_{\varphi} \log p(\varphi) = -\lambda \nabla_{\varphi} \|\varphi\|_1$ The diversity hyperparameter λ is set to 10^3.

Comparison methods. We compare GibbsDDRM with several other methods as baselines. These include MPRNet (Zamir et al., 2021) and DeblurGANv2 (Kupyn et al., 2019) as supervised learning-based baselines, pandark channel prior (Pan-DCP) (Pan et al., 2017) as an optimization-based method, and SelfDeblur (Ren et al., 2020), which utilizes deep image prior (DIP) for coestimation of the data and kernel. We also list the results for BlindDPS (Chung et al., 2023a) as reported in that paper, although it uses a prior for the blur kernel that is trained in a supervised manner, thus giving it an unfair advantage.

Evaluation metrics. For quantitative comparison of the different methods, the main metrics are the peak signal-to-noise-ratio (PSNR), the Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018), and the Fréchet Inception Distance (FID) (Heusel et al., 2017).

Results. Table 1 summarizes the quantitative results of blind image deblurring on FFHQ and AFHQ. While showing a lower FID score, which measures the quality of generated data, GibbsDDRM outperforms all the other methods in terms of the LPIPS, which measures faithfulness to the original image. To investigate the performance limit of our method, we also list the results of DDRM with a ground truth kernel.
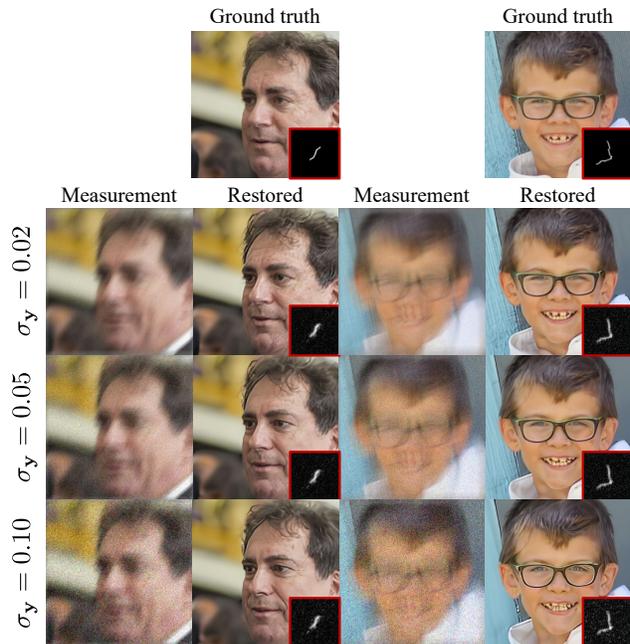

Figure 5. Blurry images and restored images obtained with a restored blur kernel in blind image deblurring under different measurement noise conditions. The top row contains the ground truth images and blur kernels.

Figure 4 visualizes the evolution of the variables for $N = 2$. We can see that even in steps where $\mathbf{x}_t$ is still quite noisy, the estimated $\mathbf{x}_{\theta,t}$ is close to the ground truth. This leads to an accurate sampling of the blur kernel, which is quite close to the ground truth at $t = 0$.

Figure 5 shows the restoration results for different measurement noise levels. We can see that even with large noise, a faithful image can be restored via the SVD.

We find that BlindDPS has a lower (better) FID score, but the restored images are relatively far from the original im-

age in terms of the quantitative results. We think that this is because our method uses DDRM, which enables efficient treatment of information obtained from measurements through the SVD, whereas BlindDPS performs more generation than is necessary for noisy observations, which may negatively affect its faithfulness.

Figure 6 shows the results obtained by our method and the comparison methods. The supervised method MPR achieves the highest PSNR of all the methods, but our method outperforms it in FID and LPIPS. It is observed that the images obtained by the MPRNet exhibit a certain degree of blurriness when compared to the ground truth images, whereas the images obtained by GibbsDDRM look to be of superior quality in terms of visual perception.

GibbsDDRM takes approximately 56 seconds of computation time per image using one RTX3090 with a batch size of 4.

4.2. Vocal dereverberation

Problem formulation.   The objective of vocal dereverberation is to restore the original dry vocal from a noisy, reverberant (wet) vocal. Appendix B gives the details of the problem formulation and the specific implementation of the GibbsDDRM that we use for this task.

Experimental settings.   The proposed method is quantitatively evaluated on wet vocal signals. A pre-trained diffusion model is trained with dry vocal signals from an internal proprietary dataset of various genres and singers, with a total duration of 15 hours. A test dataset comprising 1000 wet vocal signals, with a total duration of around 1.4 hours, is prepared by adding artificial reverb to dry vocal signals from the NHSS dataset (Sharma et al., 2021), which contains 100 English pop songs by different singers, with a total duration of 285.24 minutes. Both the training and testing data are monaural recordings sampled at 44.1 kHz. The artificial reverb is added with commercial software by using 10 presets with an RT60 shorter than 2 seconds. The wet vocal signals are prepared by creating 100×10 signals, dividing them into 5-second samples, and randomly selecting 1000 of the resulting signals.

For the GibbsDDRM algorithm, the following parameter values are used: $\eta = 0.8$, $\eta_b = 0.8$, and $\sigma_y = 1.0 \times 10^{-3}$. We set $T = 50$ for the number of sampling steps and $N = 1$. The parameter M_t is set to zero for $40 \leq t \leq 50$, and to 5 for $t \leq 40$. The linear operator's parameters are initialized using results from the WPE algorithm (Nakatani et al., 2010), which is an unsupervised method for dereverberation. The number of iterations and the learning rate for Langevin dynamics (Eq. (16)) are set to 400 and 1.0×10^{-13}, respectively. We use a Laplace prior, and the diversity hyperparameter λ is set to 2.0. Appendix C gives the details of the

Table 2. Vocal dereverberation results. **Bold**: Best.

Method	FAD ↓	SI-SDR ↑ improvement	SRMR ↑
Wet (unprocessed)	5.74	–	7.11
Reverb Conversion (Koo et al., 2021)	5.69	0.02	7.23
Music Enhancement (Kandpal et al., 2022)	7.51	−23.9	7.92
Unsupervised Dereverberation(Saito et al., 2023)	4.99	0.37	7.94
GibbsDDRM	**4.21**	**0.59**	**8.40**

network architecture and the dataset.

Comparison methods.   We evaluate the proposed method against three baselines: Reverb Conversion (RC) (Koo et al., 2021), Music Enhancement (ME) (Kandpal et al., 2022), and Unsupervised Dereverberation (UD) (Saito et al., 2023). RC is a state-of-the-art, end-to-end, DNN-based method that requires pairs of wet and dry vocal signals for dereverberation. It is trained with wet and dry vocal signals that are obtained with different commercial reverb plugins from those used for the test dataset. ME is a supervised method based on diffusion models that denoise and dereverb music signals containing vocal signals. It is trained with pairs of 16-kHz reverberant noisy and clean music signals and is evaluated at 16 kHz. UD is a method similar to ours, in that it uses DDRM; however, it differs in how it estimates the linear operator's parameters.

Evaluation metrics.   For quantitative comparison of the different methods, the metrics are the scale-invariant signal-to-distortion ratio (SI-SDR) (Roux et al., 2019) improvement, the Fréchet Audio Distance (FAD) (Kilgour et al., 2018), and the speech-to-reverberation modulation energy ratio (SRMR) (Santos et al., 2014). Because the FAD uses the pre-trained classification model VGGish (Hershey et al., 2017), which is originally trained with 16 kHz audio samples, we downsample all the signals to 16 kHz to compute FAD.

Results.   Table 2 lists the scores for each metric. GibbsD-DRM outperforms the comparison methods on all metrics. In particular, the result for UD demonstrates that our proposed way of estimating the linear operator's parameters gives better performance than UD's way. Moreover, ME doesn't work at all, which may have been because the distribution of its training dataset does not cover that of its test dataset. Indeed, the wet signals for ME's training are created using only simulated natural reverb with some background noise (Kandpal et al., 2022).

5. Conclusion

We have proposed GibbsDDRM, a method for solving blind linear inverse problems by sampling data and the parameters of a linear operator from a posterior distribution by using a PCGS. The PCGS procedure ensures that the stationary

distribution is unchanged from that of the original Gibbs sampler. GibbsDDRM performed well in experiments on blind image deblurring and vocal dereverberation, particularly in terms of preserving the original data, despite its use of a simple prior distribution for the parameters. Additionally, GibbsDDRM has problem-agnostic characteristics, which means that a single pre-trained diffusion model can be used for various tasks. One limitation of the proposed method is that it is not easily applicable to problems involving linear operators for which the SVD is computationally infeasible.

Acknowledgements

References

Anirudh, R., Thiagarajan, J. J., Kailkhura, B., and Bremer, T. An unsupervised approach to solving inverse problems using generative adversarial networks. *arXiv preprint arXiv:1805.07281*, 2018.

Candès, E. J. and Wakin, M. B. An introduction to compressive sampling. *IEEE Signal Process. Mag.*, 25(2):21–30, 2008.

Candès, E. J., Romberg, J., and Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, 52(2):489–509, 2006.

Casella, G. and George, E. I. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

Chan, T. F. and Wong, C.-K. Total variation blind deconvolution. *IEEE Trans. Image Process.*, 7(3):370–375, 1998.

Choi, J., Kim, S., Jeong, Y., Gwon, Y., and Yoon, S. ILVR: Conditioning method for denoising diffusion probabilistic models. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 14347–14356, 2021.

Choi, W., Kim, M., Chung, J., Lee, D., and Jung, S. Investigating U-Nets with various intermediate blocks for spectrogram-based singing voice separation. In *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2020a.

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. StarGAN v2: Diverse image synthesis for multiple domains. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8188–8197, 2020b.

Chung, H., Kim, J., Kim, S., and Ye, J. C. Parallel diffusion models of operator and image for blind inverse problems. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023a.

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

Dhariwal, P. and Nichol, A. Diffusion models beat GANs on image synthesis. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 8780–8794, 2021.

Eaton, J., Gaubitch, N. D., Moore, A. H., and Naylor, P. A. The ace challenge — corpus description and performance evaluation. In *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5, 2015. doi: 10.1109/WASPAA.2015.7336912.

Fazel, M., Candes, E., Recht, B., and Parrilo, P. Compressed sensing and robust recovery of low rank matrices. In *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pp. 1043–1047. IEEE, 2008.

Gao, X., Sitharam, M., and Roitberg, A. E. Bounds on the Jensen gap, and implications for mean-concentrated distributions. *arXiv preprint arXiv:1712.05267*, 2017.

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., and Wilson, K. CNN architectures for large-scale audio classification. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 131–135, 2017.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6629–6640, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 33:6840–6851, 2020.

K. A. Reddy, C., Dubey, H., Koishida, K., Asokan Nair, A., Gopal, V., Cutler, R., Braun, S., Gamper, H., Aichner, R., and Srinivasan, S. Interspeech 2021 deep noise suppression challenge. In *Interspeech*, 2021.

Kadkhodaie, Z. and Simoncelli, E. P. Solving linear inverse problems using the prior implicit in a denoiser. In *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*, 2020.

Kail, G., Tourneret, J.-Y., Hlawatsch, F., and Dobigeon, N. Blind deconvolution of sparse pulse sequences under a minimum distance constraint: A partially collapsed Gibbs sampler method. *IEEE Trans. Signal Process.*, 60 (6):2727–2743, 2012.

Kandpal, N., Nieto, O., and Jin, Z. Music enhancement via image translation and vocoding. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 3124–3128. IEEE, 2022.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4401–4410, 2019.

Kawar, B., Vaksman, G., and Elad, M. SNIPS: Solving noisy inverse problems stochastically. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 21757–21769, 2021.

Kawar, B., Elad, M., Ermon, S., and Song, J. Denoising diffusion restoration models. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Kilgour, K., Zuluaga, M., Roblek, D., and Sharifi, M. Fréchet Audio Distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

Koo, J., Paik, S., and Lee, K. Reverb conversion of mixed vocal tracks using an end-to-end convolutional deep neural network. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 81–85. IEEE, 2021.

Krishnan, D. and Fergus, R. Fast image deconvolution using hyper-Laplacian priors. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 22, 2009.

Kruse, J., Rother, C., and Schmidt, U. Learning to push the limits of efficient FFT-based image deconvolution. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 4586–4594, 2017.

Kupyn, O., Martyniuk, T., Wu, J., and Wang, Z. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 8878–8887, 2019.

Lai, C.-H., Takida, Y., Murata, N., Uesaka, T., Mitsufuji, Y., and Ermon, S. Improving score-based diffusion models by enforcing the underlying score Fokker-Planck equation. 2022.

Langevin, P. *On the theory of Brownian motion.* 1908.

Larsen, E. and Aarts, R. M. *Audio bandwidth extension: application of psychoacoustics, signal processing and loudspeaker design.* John Wiley & Sons, 2005.

Larsson, G., Maire, M., and Shakhnarovich, G. Learning representations for automatic colorization. In *European conference on computer vision*, pp. 577–593. Springer, 2016.

Liu, J. S., Wong, W. H., and Kong, A. Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81 (1):27–40, 1994.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *Proc. International Conference on Learning Representation (ICLR)*, 2019.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. Mixed precision training. In *Proc. International Conference on Learning Representation (ICLR)*, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *Proc. International Conference on Learning Representation (ICLR)*, 2018.

Nakatani, T., Yoshioka, T., Kinoshita, K., Miyoshi, M., and Juang, B.-H. Speech dereverberation based on variance-normalized delayed linear prediction. *IEEE Trans. Audio, Speech, Lang. Process.*, 18(7):1717–1731, 2010.

Nichol, A. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *Proc. International Conference on Machine Learning (ICML)*, pp. 8162–8171. PMLR, 2021.

Pan, J., Sun, D., Pfister, H., and Yang, M.-H. Blind image deblurring using dark channel prior. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1628–1636, 2016.

Pan, J., Sun, D., Pfister, H., and Yang, M.-H. Deblurring images via dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2315–2328, 2017.

Parmar, G., Zhang, R., and Zhu, J.-Y. On aliased resizing and surprising subtleties in gan evaluation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11410–11420, 2022.

Ren, D., Zhang, K., Wang, Q., Hu, Q., and Zuo, W. Neural blind deconvolution using deep priors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3341–3350, 2020.

Rick Chang, J., Li, C.-L., Poczos, B., Vijaya Kumar, B., and Sankaranarayanan, A. C. One network to solve them all–solving linear inverse problems using deep projection models. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 5888–5897, 2017.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, 2015.

Roux, J. L., Wisdom, S., Erdogan, H., and Hershey, J. R. SDR – Half-baked or well done? In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 626–630, 2019. doi: 10.1109/ICASSP.2019.8683855.

Saito, K., Murata, N., Uesaka, T., Lai, C.-H., Takida, Y., Fukui, T., and Mitsufuji, Y. Unsupervised vocal dereverberation with diffusion-based generative models. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2023.

Santos, J. F., Senoussaoui, M., and Falk, T. H. An improved non-intrusive intelligibility metric for noisy and reverberant speech. In *Proc. Int. Workshop Acoust. Signal Enhancement (IWAENC)*, pp. 55–59, 2014. doi: 10.1109/IWAENC.2014.6953337.

Sedghi, H., Gupta, V., and Long, P. M. The singular values of convolutional layers. In *Proc. International Conference on Learning Representation (ICLR)*, 2019.

Sharma, B., Gao, X., Vijayan, K., Tian, X., and Li, H. NHSS: A speech and singing parallel database. *Speech Communication*, 133:9–22, 2021.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. International Conference on Machine Learning (ICML)*, pp. 2256–2265. PMLR, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 32: 11895–11907, 2019.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 12438–12448, 2020.

Song, Y., Shen, L., Xing, L., and Ermon, S. Solving inverse problems in medical imaging with score-based generative models. In *NeurIPS 2021 Workshop on Deep Learning and Inverse Problems*, 2021a.

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

Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., and Li, Y. MAXIM: Multi-axis MLP for image processing. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5769–5780, 2022.

Van Dyk, D. A. and Park, T. Partially collapsed Gibbs samplers: Theory and methods. *Journal of the American Statistical Association*, 103(482):790–796, 2008.

Whang, J., Lei, Q., and Dimakis, A. Solving inverse problems with a flow-based noise model. In *Proc. International Conference on Machine Learning (ICML)*, pp. 11146–11157. PMLR, 2021.

Xu, L., Zheng, S., and Jia, J. Unnatural l0 sparse representation for natural image deblurring. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1107–1114, 2013.

Yeh, R. A., Chen, C., Yian Lim, T., Schwing, A. G., Hasegawa-Johnson, M., and Do, M. N. Semantic image inpainting with deep generative models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5485–5493, 2017.

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M.-H., and Shao, L. Multi-stage progressive image restoration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14821–14831, 2021.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Zhu, B., Liu, J. Z., Cauley, S. F., Rosen, B. R., and Rosen, M. S. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492, 2018.

A. Proofs

A.1. Proof of Proposition 3.1

Proposition 3.1 *The PCGS defined in Algorithm 1 has the true posterior distribution $p(\mathbf{x}_{0:T}, \boldsymbol{\varphi}|\mathbf{y})$ as its stationary distribution if the approximations to the conditional distributions are exact.*

Before giving the proof, we revisit three basic tools for constructing a partially collapsed Gibbs sampler (PCGS) (Van Dyk & Park, 2008).

Gibbs sampler. Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_J)^\mathsf{T}$ be a vector of J variables, and let $\boldsymbol{\theta}_{\tilde{j}}$ denote $\boldsymbol{\theta}$ without the j th element θ_j. To obtain samples from $p(\boldsymbol{\theta})$, a Gibbs sampler (Casella & George, 1992) iteratively generates samples of each θ_j from $p(\theta_j|\boldsymbol{\theta}_{\tilde{j}})$ in an arbitrary order. The generated samples approximate the joint distribution of all variables.

PCGS. A PCGS is an extension of the Gibbs sampler that facilitates the following three basic tools (see (Van Dyk & Park, 2008) for details).

- *Marginalization.* Rather than sampling only θ_j in a step, other variables may be sampled with θ_j instead of being conditioned on. This process is called marginalization, and it can improve the convergence rate significantly, especially with a strong correlation between the target variables. Within an entire PCGS iteration, certain parameters can be sampled in more than one step.

- *Trimming.* If a variable is sampled in several steps and is not used as a condition on these steps, only the value sampled in the last step is relevant because the other values are never used. Such unused variables can thus be removed from the respective sampling distribution. This reduces the complexity of the sampling steps without affecting the convergence behavior.

- *Permutation.* It is reasonable to choose an (arbitrary) sampling order such that trimming can be performed. After trimming, permutations are only allowed if they preserve the justification of the trimming that has already been applied.

For example, the following PCGS for sampling $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W})$ is a simple PCGS.

$$
\begin{aligned}
&\text{Step 1. Sample } \mathbf{Y} \text{ from } p(\mathbf{Y}, \mathbf{\cancel{W}}|\mathbf{X}, \mathbf{Z}) \\
&\text{Step 2. Sample } \mathbf{Z} \text{ from } p(\mathbf{Z}, \mathbf{\cancel{W}}|\mathbf{X}, \mathbf{Y}) \\
&\text{Step 3. Sample } \mathbf{W} \text{ from } p(\mathbf{W}|\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \\
&\text{Step 4. Sample } \mathbf{X} \text{ from } p(\mathbf{X}|\mathbf{W}, \mathbf{Y}, \mathbf{Z})
\end{aligned}
\tag{18}
$$

Here, the random variable $\mathbf{W}$ is trimmed in steps 1 and 2 because it is sampled in step 3 before being included in the conditional variables. Note that the order of steps 3 and 4 cannot be interchanged. The reason is that the variable $\mathbf{W}$, which is trimmed in steps 1 and 2, would be included among the conditional variables in $p(\mathbf{X}|\mathbf{W}, \mathbf{Y}, \mathbf{Z})$, thus altering the sampler's stationary distribution.

Proof. To show that the proposed sample is a valid PCGS, we transform a naïve Gibbs sampler by applying the above PCGS tools to the proposed PCGS. First, we consider the naïve Gibbs sampler defined in Algorithm 2, which we denote as Sampler 1.

Sampler 1 has a stationary distribution $p(\mathbf{x}_{0:T}, \boldsymbol{\varphi}|\mathbf{y})$, since it is a naïve Gibbs sampler for the joint distribution in Eq. (7). In Gibbs sampling, the stationary distribution is unaffected by repeating certain steps and changing the order of the steps. Therefore, the sampling scheme depicted in Figure 3 constructs Sampler 2, which is defined in Algorithm 3.

Next, Sampler 2 is converted to Sampler 3, which is defined in Algorithm 4, by marginalizing the variables ψ_t. Subsequently, we convert the Sampler 3 to our proposed sampler by employing the PCGS trimming operation and approximating the conditional distributions. The variables ψ_t for $t = 0, 1, \ldots, T$ can be trimmed from Sampler 3, as they do not appear in the conditional variables of the conditional distribution before they are next sampled. Because the proposed PCGS corresponds to a sampler that omits ψ_t from Sampler 3, it has the true posterior distribution $p(\mathbf{x}_{0:T}|\mathbf{y})$ as its stationary distribution. Thus, if the approximations for the conditional distributions are exact, the PCGS has the true posterior distribution as its stationary distribution. □

Algorithm 2 Sampler 1 for the posterior in Eq. (7)

Input: Measurement $\mathbf{y}$, initial values $\boldsymbol{\varphi}^{(0)}$, $\mathbf{x}_{0:T}^{(0)}$.
Output: Restored data $\mathbf{x}_0^{(N)}$, linear operator's parameters $\boldsymbol{\varphi}^{(N)}$
for $n = 1$ **to** N **do**
    Sample $\mathbf{x}_T^{(n)} \sim p(\mathbf{x}_T | \mathbf{x}_{0:T-1}^{(n-1)}, \boldsymbol{\varphi}^{(n-1)}, \mathbf{y})$
    for $t = T - 1$ **to** 0 **do**
        Sample $\mathbf{x}_t^{(n)} \sim p(\mathbf{x}_t | \mathbf{x}_{0:t-1}^{(n-1)}, \mathbf{x}_{t+1:T}^{(n)}, \boldsymbol{\varphi}^{(n-1)}, \mathbf{y})$
    end for
    $\boldsymbol{\varphi}^{(n)} \sim p(\boldsymbol{\varphi} | \mathbf{x}_{0:T}^{(n)}, \mathbf{y})$
end for

Algorithm 3 Sampler 2 for the posterior in Eq. (7)

Input: Measurement $\mathbf{y}$, initial values $\boldsymbol{\varphi}^{(0,0)}$, $\mathbf{x}_{0:T}^{(0,M_0)}$
Output: Restored data $\mathbf{x}_0^{(N,M_0)}$, parameters of linear operator $\boldsymbol{\varphi}^{(N,K)}$
$K \leftarrow 0$ // K counts the number of updates for φ in a cycle.
for $n = 1$ **to** N **do**
    $\boldsymbol{\varphi}^{(n,0)} \leftarrow \boldsymbol{\varphi}^{(n-1,K)}$
    $K \leftarrow 0$
    $\psi_T \leftarrow \{\mathbf{x}_0^{(n-1,M_0)}, \mathbf{x}_1^{(n-1,M_1)}, \ldots, \mathbf{x}_t^{(n-1,M_t)}, \ldots, \mathbf{x}_{T-1}^{(n-1,M_{T-1})}\}$
    Sample $\mathbf{x}_T^{(n,0)} \sim p(\mathbf{x}_T | \boldsymbol{\varphi}^{(n,K)}, \underset{\sim}{\psi_T}, \mathbf{y})$
    for $t = T - 1$ **to** 0 **do**
        $\psi_t \leftarrow \{\mathbf{x}_0^{(n-1,M_0)}, \mathbf{x}_1^{(n-1,M_1)}, \ldots, \mathbf{x}_{t-1}^{(n-1,M_{t-1})}\}$
        $\chi_t \leftarrow \{\mathbf{x}_{t+1}^{(n,M_{t+1})}, \mathbf{x}_{t+2}^{(n,M_{t+2})}, \cdots, \mathbf{x}_T^{(n,0)}\}$
        Sample $\mathbf{x}_t^{(n,0)} \sim p(\mathbf{x}_t | \boldsymbol{\varphi}^{(n,K)}, \underset{\sim}{\psi_t}, \chi_t, \mathbf{y})$
        for $m = 1$ **to** M_t **do**
            Sample $\boldsymbol{\varphi}^{(n,K+1)} \sim p(\boldsymbol{\varphi} | \mathbf{x}_t^{(n,m-1)}, \underset{\sim}{\psi_t}, \chi_t, \mathbf{y})$
            $K \leftarrow K + 1$
            Sample $\mathbf{x}_t^{(n,m)} \sim p(\mathbf{x}_t | \boldsymbol{\varphi}^{(n,K)}, \underset{\sim}{\psi_t}, \chi_t, \mathbf{y})$
        end for
    end for
end for

A.2. Proof of Theorem 3.2

We follow the result from (Chung et al., 2023b;a). First, we confirm the following lemmas.

Lemma A.1. *Let $\phi(\cdot)$ be a univariate Gaussian density function with mean μ and variance σ^2. $\phi(\cdot)$ is L-Lipschitz such that $\forall x_1, x_2 \in \mathbb{R}$,*

$$|\phi(x_1) - \phi(x_2)| \leq L|x_1 - x_2|, \tag{19}$$

where $L = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-1/2}$.

Proof. Since $\phi(\cdot)$ is an everywhere differentiable function and it has the bounded first derivative, we use the mean value theorem to get

$$\forall x_1, x_2 \in \mathbb{R}, |\phi(x_1) - \phi(x_2)| \leq \|\phi'\|_\infty |x_1 - x_2|. \tag{20}$$

Since L is the minimal value for Eq. (19), we have that $L \leq \|\phi'\|_\infty$. Taking the limit $x_2 \to x_1$ gives $|\phi'(x)| \leq L$, and thus

---

Algorithm 4 Sampler 3 for the posterior in Eq. (7)

Input: Measurement $\mathbf{y}$, initial values $\boldsymbol{\varphi}^{(0,0)}$, $\mathbf{x}_{0:T}^{(0,M_0)}$
Output: Restored data $\mathbf{x}_0^{(N,M_0)}$, parameters of linear operator $\boldsymbol{\varphi}^{(N,K)}$
$K \leftarrow 0$ // K counts the number of updates for $\boldsymbol{\varphi}$ in a cycle.
for $n = 1$ **to** N **do**
   $\boldsymbol{\varphi}^{(n,0)} \leftarrow \boldsymbol{\varphi}^{(n-1,K)}$
   $K \leftarrow 0$
   $\psi_T \leftarrow \{\mathbf{x}_0^{(n-1,M_0)}, \mathbf{x}_1^{(n-1,M_1)}, \dots, \mathbf{x}_t^{(n-1,M_t)}, \dots, \mathbf{x}_{T-1}^{(n-1,M_{T-1})}\}$
   Sample $\{\mathbf{x}_T^{(n,0)}, \underset{\sim}{\psi_T}\} \sim p(\mathbf{x}_T | \boldsymbol{\varphi}^{(n,K)}, \mathbf{y})$
   for $t = T - 1$ **to** 0 **do**
     $\psi_t \leftarrow \{\mathbf{x}_0^{(n-1,M_0)}, \mathbf{x}_1^{(n-1,M_1)}, \dots, \mathbf{x}_{t-1}^{(n-1,M_{t-1})}\}$
     $\chi_t \leftarrow \{\mathbf{x}_{t+1}^{(n,M_{t+1})}, \mathbf{x}_{t+2}^{(n,M_{t+2})}, \cdots, \mathbf{x}_T^{(n,0)}\}$
     Sample $\{\mathbf{x}_t^{(n,0)}, \underset{\sim}{\psi_t}\} \sim p(\mathbf{x}_t | \boldsymbol{\varphi}^{(n,K)}, \chi_t, \mathbf{y})$
     for $m = 1$ **to** M_t **do**
       Sample $\{\boldsymbol{\varphi}^{(n,K+1)}, \underset{\sim}{\psi_t}\} \sim p(\boldsymbol{\varphi} | \mathbf{x}_t^{(n,m-1)}, \chi_t, \mathbf{y})$
       $K \leftarrow K + 1$
       Sample $\{\mathbf{x}_t^{(n,m)}, \underset{\sim}{\psi_t}\} \sim p(\mathbf{x}_t | \boldsymbol{\varphi}^{(n,K)}, \chi_t, \mathbf{y})$
     end for
   end for
end for

$\|\phi'\|_\infty \leq L$. Hence

$$L = \|\phi'\|_\infty = \left\| -\frac{x - \mu}{\sigma^2} \phi(x) \right\|_\infty. \tag{21}$$

Since the derivative of ϕ' is given as

$$\phi''(x) = -\sigma^{-2}(1 - \sigma^{-2}(x - \mu)^2)\phi(x), \tag{22}$$

and the maximum is attained when $x = \mu \pm \sigma$, we have

$$L = \|\phi'\|_\infty = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-1/2} \tag{23}$$

$\square$

Lemma A.2. *Let $f(\cdot)$ be an isotropic multivariate Gaussian density function with mean $\boldsymbol{\mu}$ and variance $\sigma^2 \mathbf{I}$. $f(\cdot)$ is L-Lipschitz such that $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$,*

$$\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|, \tag{24}$$

where

$$L = \frac{1}{\sigma\left(\sqrt{2\pi\sigma^2}\right)^d} e^{-1/2} \tag{25}$$

Proof. We first evaluate the value of $\max_{\mathbf{x}} \|\nabla f(\mathbf{x})\|$, where $f(\mathbf{x}) = \prod_{i=1}^d \phi(x_i)$. Without loss of generality, we assume

$\mu = 0$.

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \ldots, \frac{\partial f(\mathbf{x})}{\partial x_d}\right]^\mathsf{T}$$

$$= \left[\phi'(x_1)\prod_{i \neq 1}\phi(x_i), \ldots, \phi'(x_d)\prod_{i \neq d}\phi(x_i)\right]^\mathsf{T}$$

$$= \left[-\frac{x_1}{\sigma^2}\prod_{i=1}^{d}\phi(x_i), \ldots, -\frac{x_d}{\sigma^2}\prod_{i=1}^{d}\phi(x_i)\right]^\mathsf{T}$$

$$= -\frac{\prod_{i=1}^{d}\phi(x_i)}{\sigma^2}[x_1, \ldots, x_d]^\mathsf{T}. \tag{26}$$

Therefore, $\max_{\mathbf{x}}\|\nabla f(\mathbf{x})\|$ can be evaluated as follows,

$$\|\nabla f(\mathbf{x})\| = \sqrt{x_1^2 + \cdots + x_d^2}\frac{\prod_{i=1}^{d}\phi(x_i)}{\sigma^2}$$

$$= \sqrt{x_1^2 + \cdots + x_d^2}\frac{\exp\left(-\frac{x_1^2 + \cdots + x_d^2}{2\sigma^2}\right)}{\sigma^2 \cdot \left(\sqrt{2\pi\sigma^2}\right)^d}$$

$$= r\frac{\exp(-\frac{r^2}{2\sigma^2})}{\sigma^2 \cdot \left(\sqrt{2\pi\sigma^2}\right)^d}, \quad r \geq 0 \quad \left(r = \sqrt{x_1^2 + \cdots + x_d^2}\right)$$

$$\overset{(a)}{\leq} \frac{1}{\sigma\left(\sqrt{2\pi\sigma^2}\right)^d}e^{-1/2} = C_{\text{multi}}, \tag{27}$$

where the equality holds when $r \,(= \sqrt{x_1^2 + \cdots + x_d^2}) = \sigma$. (a) is by the result of the lemma A.1. Here, by the mean value theorem, for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$, the following holds:

$$\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\| \leq C_{\text{multi}}\|\mathbf{x}_1 - \mathbf{x}_2\|. \tag{28}$$

By setting $\mathbf{x}_1 = [\sigma, 0, \ldots, 0]^\mathsf{T}$ and taking the limit $\mathbf{x}_2 \to \mathbf{x}_1$, the equality holds. Hence, $f(\cdot)$ is L-Lipschitz with the Lipschitz constant $L = C_{\text{multi}}$. □

Lemma A.3. *Let $\mathbf{H} \in \mathbb{R}^{d_\mathbf{y} \times d_\mathbf{x}}$ be a linear operator. The linear operator is L-Lipschitz such that $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{d_\mathbf{x}}$,*

$$\|\mathbf{H}\mathbf{x}_1 - \mathbf{H}\mathbf{x}_2\| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|, \tag{29}$$

where L is the largest singular value of $\mathbf{H}$.

This property has been reported in several papers, such as (Miyato et al., 2018).

Theorem 3.2 *(modified version of Theorem 1 in (Chung et al., 2023b)) For the measurement model in Eq. (1), we have*

$$p(\mathbf{y}|\mathbf{x}_{t:T}, \boldsymbol{\varphi}) \simeq p(\mathbf{y}|\mathbf{x}_{\theta,t}, \boldsymbol{\varphi}), \tag{30}$$

and the approximation error can be quantified with the Jensen gap (Gao et al., 2017), which is upper bounded by

$$\mathcal{J} \leq \frac{1}{\sigma_\mathbf{y}\left(\sqrt{2\pi\sigma_\mathbf{y}^2}\right)^{d_\mathbf{y}}}e^{-1/2}s_1 m_1, \tag{31}$$

where $m_1 := \int \|\mathbf{x}_0 - \mathbf{x}_{\theta,t}\|p(\mathbf{x}_0|\mathbf{x}_{t:T})d\mathbf{x}_0$, and s_1 is the largest singular value of $\mathbf{H}_\boldsymbol{\varphi}$.

Proof. In our case, the Jensen gap (Gao et al., 2017) is defined as follows:

$$\mathcal{J} = |p(\mathbf{y}|\mathbf{x}_{t:T}, \boldsymbol{\varphi}) - p(\mathbf{y}|\mathbf{x}_{\theta,t}, \boldsymbol{\varphi})|.$$

Let $f(\boldsymbol{\mu})$ be an isotropic multivariate Gaussian density function with mean $\boldsymbol{\mu}$ and variance $\sigma_{\mathbf{y}}^2 \mathbf{I}$, and thus $p(\mathbf{y}|\mathbf{x}_0, \boldsymbol{\varphi}) = f(\mathbf{H}_{\boldsymbol{\varphi}}\mathbf{x}_0)$ in our case. The Jensen gap is evaluated as follows:

$$
\begin{aligned}
\mathcal{J} &= |p(\mathbf{y}|\mathbf{x}_{t:T}, \boldsymbol{\varphi}) - p(\mathbf{y}|\mathbf{x}_{\theta,t}, \boldsymbol{\varphi})| \\
&= \left| \int (p(\mathbf{y}|\mathbf{x}_{t:T}, \mathbf{x}_0, \boldsymbol{\varphi}) - p(\mathbf{y}|\mathbf{x}_{\theta,t}, \boldsymbol{\varphi})) \, p(\mathbf{x}_0|\mathbf{x}_{t:T}) d\mathbf{x}_0 \right| \\
&\overset{(a)}{=} \left| \int (p(\mathbf{y}|\mathbf{x}_0, \boldsymbol{\varphi}) - p(\mathbf{y}|\mathbf{x}_{\theta,t}, \boldsymbol{\varphi})) \, p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right| \\
&= \left| \int (f(\mathbf{H}_{\boldsymbol{\varphi}}\mathbf{x}_0) - f(\mathbf{H}_{\boldsymbol{\varphi}}\mathbf{x}_{\theta,t})) \, p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \right| \\
&\overset{(b)}{\leq} \frac{1}{\sigma_{\mathbf{y}} \left(\sqrt{2\pi\sigma_{\mathbf{y}}^2} \right)^{d_{\mathbf{y}}}} e^{-1/2} \int \|\mathbf{H}_{\boldsymbol{\varphi}}\mathbf{x}_0 - \mathbf{H}_{\boldsymbol{\varphi}}\mathbf{x}_{\theta,t}\| p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \\
&\overset{(c)}{\leq} \frac{1}{\sigma_{\mathbf{y}} \left(\sqrt{2\pi\sigma_{\mathbf{y}}^2} \right)^{d_{\mathbf{y}}}} e^{-1/2} s_1 \int \|\mathbf{x}_0 - \mathbf{x}_{\theta,t}\| p(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0 \\
&= \frac{1}{\sigma_{\mathbf{y}} \left(\sqrt{2\pi\sigma_{\mathbf{y}}^2} \right)^{d_{\mathbf{y}}}} e^{-1/2} s_1 m_1,
\end{aligned}
\tag{32}
$$

where (a) is by the conditional independence of $\mathbf{y}$ and $\mathbf{x}_{t:T}$ given $\mathbf{x}_0$ and the Markov property of $\mathbf{x}_{t:T}$, and (b) and (c) are by the lemmas A.2 and A.3. □

B. Instantiation of blind linear inverse problems

Blind image deblurring. The aim of blind image deblurring is to restore a clean image from a noisy blurred image without knowledge of the blur kernel. The problem is formulated as follows:

$$\mathbf{y} = \mathbf{k} * \mathbf{x}_0 + \mathbf{z}, \tag{33}$$

where $\mathbf{k}$ is the blur kernel, corresponding to the parameters $\boldsymbol{\varphi}$ in our setting, and $*$ denotes the convolution operator. Although dealing with this problem in our framework requires the SVD of the convolution operator, it can be computed efficiently by using an FFT (Sedghi et al., 2019; Kruse et al., 2017). Thus, the SVD enables efficient calculation in the spectral domain. In performing the SVD with an FFT, it is necessary to consider signals in the complex domain; however, the proposed method can be naturally extended to the complex case.

Vocal dereverberation. The details of dealing with vocal dereverberation as a linear inverse problem are discussed in (Saito et al., 2023). Let $y_{\tau,f}^{\text{wet}} \in \mathbb{C}$ be the wet (reverberant) vocal signals in a short-time Fourier transform (STFT) domain, where τ and f denote the respective time and frequency indices. We use the following measurement model:

$$y_{\tau,f}^{\text{wet}} = \sum_{l=0}^{L-1} g_{l,f}^* x_{\tau-l,f}^{\text{dry}} + z_{\tau,f}, \tag{34}$$

where $x_{\tau,f}^{\text{dry}} \in \mathbb{C}$ and $g_{\tau,f} \in \mathbb{C}$ are the dry vocal signals and the acoustic transfer function between wet and dry signals, respectively. Here, we assume additive noise $z_{\tau,f} \in \mathbb{C}$. $(\cdot)^*$ denotes the complex conjugate, and L is the length of reverberation. As with blind image deblurring, the linear operator, in this case, is a convolution operator whose acoustic transfer function is unknown. Thus, the efficient method of performing the SVD by using an FFT is applicable.

C. Details on experimental settings

C.1. Blind image deblurring.

Comparison methods. For methods requiring training data, images from the dataset are corrupted with blur kernels that are generated by using the MotionBlur library[3] and Gaussian noise with variance $\sigma_{\mathbf{y}} = 0.02$ is added. The blur kernel size is 64×64, and the intensity value is determined for each kernel by uniform sampling from the range [0.4, 0.6].

MPRNet ([Zamir et al., 2021]). We use the official implementation [4] for the deblurring task, with the recommended parameters, learning rate decay, and neural network architectures. The model is trained for 100k iterations with a batch size of 4 for both the FFHQ and AFHQ datasets.

DeblurGANv2 ([Kupyn et al., 2019]). We use the official implementation [5] while adhering to the default settings for the parameters and network architectures. Specifically, the model is trained by minimizing the sum of the pixel distance loss, WGAN-gp adversarial loss, and perceptual loss with the weight parameters specified in the official implementation. The generator uses Inception-ResNet-v2 as its backbone. For both the FFHQ and AFHQ datasets, the model is trained for 500k iterations with a batch size of 1. The hyperparameters for the loss are set to $\lambda_{\mathrm{pixel}} = 5.0 \times 10^{-1}$, $\lambda_{\mathrm{adv}} = 6.0 \times 10^{-3}$, and $\lambda_{\mathrm{perceptual}} = 1.0 \times 10^{-2}$.

Pan-DCP ([Pan et al., 2016]). We use the official implementation [6] with the parameters recommended for facial images. For the hyperparameters, we use $\lambda_{\mathrm{dark}} = 4.0 \times 10^{-3}$, and $\lambda_{\mathrm{grad}} = 4.0 \times 10^{-3}$. The number of iterations is set to 5.

SelfDeblur ([Ren et al., 2020]). We use the official implementation [7] with the default settings for YCbCR and a fixed learning rate of 0.01 for 2500 steps. The optimization process involves minimizing the mean squared error (MSE) for the initial 500 steps, followed by a switch to the structural similarity index (SSIM) loss function for the remaining steps.

Details on evaluation metrics. The FID scores reported in the paper are calculated using the cleanfid library ([Parmar et al., 2022]) [8]. Specifically, for FFHQ, the evaluation is conducted with 1,000 restored images and 70,000 images from the training and validation set. Similarly, for AFHQ, the evaluation is conducted on 500 restored images and 4,739 images from the training set. The limited number of samples used in the evaluation is due to the computational complexity of the proposed method. The BlindDPS paper doesn't provide details on the calculation of FID, so there may be slight differences in the reported values.

C.2. Vocal dereverberation.

The pre-trained diffusion model for GibbsDDRM is trained with only dry vocal signals from an internal dataset containing various genres of songs by various singers. The total signal duration is around 15 hours. For a test dataset, we use 1000 wet vocal signals (1.4 hours in total) by adding artificial reverb to dry vocal signals from another dataset, the NHSS dataset ([Sharma et al., 2021]). That dataset contains 100 English pop songs (20 unique songs) by different singers, with a total signal duration of 285.24 minutes. Each song for training and testing is sampled at 44.1 kHz and features monaural recording. For artificial reverb, we use the presets for vocals in the FabFilter Pro-R plug-in [9], which is a commercial artificial reverb plug-ins. From a total of 19 kinds of vocal reverb presets, we use all the presets whose RT60 is shorter than 2 seconds (10 in total). We prepare wet test dataset by creating 100×10 signals, dividing them into 5-second samples, and randomly selecting 1000 of the resulting signals.

The implementation of our method and the network architecture of the pre-trained diffusion model are mostly based on the code provided by the authors of the DDRM paper [10]. We slightly modify certain parts as follows. We convert each audio input to a complex-valued STFT representation by using a window size of 1024, a hop size of 256, and a Hann window. Further, to follow the original input configuration, we cut the direct-current components of the input signals and input them as 2-channeled 512×512 image data. The first and second channels correspond to the respective real and imaginary parts.

---

[3]https://github.com/LeviBorodenko/motionblur
[4]https://github.com/swz30/MPRNet
[5]https://github.com/VITA-Group/DeblurGANv2
[6]https://jspan.github.io/projects/dark-channel-deblur
[7]https://github.com/csdwren/SelfDeblur
[8]https://github.com/GaParmar/clean-fid
[9]https://www.fabfilter.com/products/pro-r-reverb-plug-in
[10]https://github.com/bahjat-kawar/ddrm

We modify the original U-Net (Ronneberger et al., 2015) architecture of the pre-trained model used on DDRM by adding a time-distributed, fully connected (TFC) layer (Choi et al., 2020a) to the last layer of every residual block expecting the TFC layers to capture the harmonic structure of music signals efficiently.

For the training, we reduce the diffusion model's size by having fewer trainable parameters (31.3 M), and the training took less than three days with an NVIDIA A100 GPU. The hyperparameters for the training of the diffusion model are in Table 3. We also incorporate an adaptive group normalization (Dhariwal & Nichol, 2021) into each residual block. We train the model using AdamW (Loshchilov & Hutter, 2019) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ in 16-bit precision (Micikevicius et al., 2018). We use an exponential moving average over model parameters with a rate of 0.9999 (Song & Ermon, 2020).

Table 3. Hyperparameters for training diffusion model on dry vocal signals. We follow the same notations defined in (Dhariwal & Nichol, 2021)

Diffusion steps	4000
Noise schedule	cosine (Nichol & Dhariwal, 2021)
Model size	31.3 M
Channels	64
Depth	2
Channels multiple	$1, 1, 2, 2, 4, 4$
Heads	2
Attention resolution	$32, 16$
BigGAN up/downsample	✓
Dropout	0.0
Batch size	6
Iterations	370K
Learning rate	1.0×10^{-4}

For initialization of the linear operator, we used the WPE with the parameters $L = 150$, $D = 4$, and one iteration. GibbsDDRM takes 36 seconds to restore 1 second vocal signals, whereas UD takes 6 seconds.

Comparison methods. **Reverb conversion**: A state-of-the-art end-to-end DNN-based method for vocal dereverberation. We use the original code and the pre-trained model[11], which is trained with the pairs of 44.1 kHz wet and dry vocal signals. Note that the wet signals are reverbed with the artificial reverb for vocal taken from the different commercial reverb plug-ins (e.g., [12][13]) from those of our test dataset (Koo et al., 2021). We input pairs of wet and dry signals since this method needs them for dereverberation.

Music enhancement: A supervised method to denoise and dereverb music signals based on diffusion models (Kandpal et al., 2022). We use both the original code and the pre-trained model specified in the paper. Since ME is trained with pairs of 16 kHz reverberant noisy and clean music signals containing vocal signals, we evaluate this method at 16 kHz for all the objective metrics. Note that the wet signals of the training dataset are created using room impulse responses from the DNS Challenge dataset (K. A. Reddy et al., 2021), which may have some different characteristics from artificial reverb for vocal signals, and adding the background noise from the ACE Challenge dataset (Eaton et al., 2015).

UnsupervisedDereverb: An unsupervised method for vocal dereverberation (Saito et al., 2023). This method is similar to our GibbsDDRM, which utilizes DDRM. However, it differs in how it estimates the linear operator's parameter. We use the same pre-trained diffusion model as GibbsDDRM. We set $L = 150, D = 4$, the number of iterations of WPE to one, $\eta = 0.8, \eta_b = 0.8, \sigma_y = 1.0 \times 10^{-3}$, with the number of sampling steps T set to 50. The number of iterations, the learning rate, and the regularization parameter for refinement of the linear operator are set to 10000, 1.0×10^{-6}, and 1.0, respectively.

---

[11]The original code and the pre-trained model are shared by Junghyun Koo from the Department of Intelligence and Information at Seoul National University. Mr. Koo also assisted with the discussion of the RC results of our experiment.

[12]https://valhalladsp.com/shop/reverb/valhalla-room/

[13]https://valhalladsp.com/shop/reverb/valhalla-vintage-verb/

D. Additional Results.
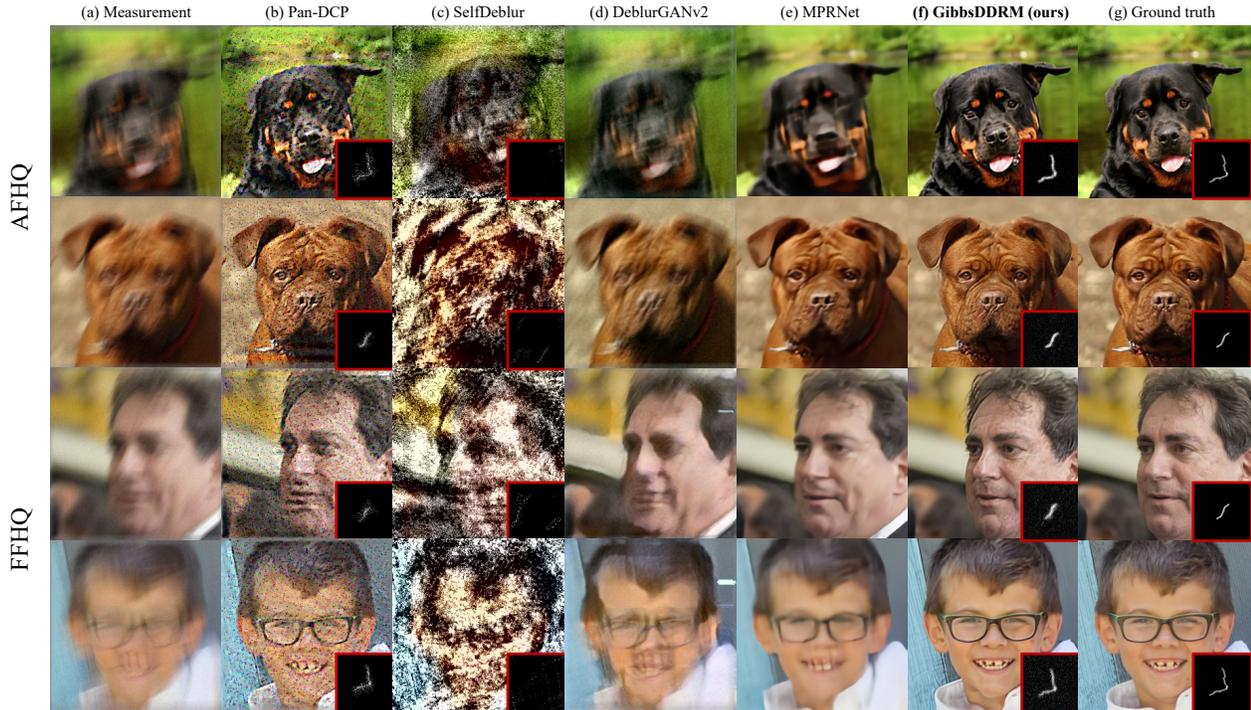
D.1. Blind image deblurring.

Figure 6. Blind image deblurring results on the FFHQ and AFHQ datasets: (a) measurements, (b) Pan-DCP (Pan et al., 2016), (c) SelfDeblur (Ren et al., 2020), (d) DeblurGANv2 (Kupyn et al., 2019), (e) MPRNet (Zamir et al., 2021), (f) GibbsDDRM (ours), and (g) ground truth. The kernels are also shown for methods that estimate them.

Qualitative comparison. We show the results of our method and comparison methods in Figure. 6. The images estimated by GibbsDDRM appear perceptually similar to the ground truth images, but the images estimated by MPRNet have better quality in terms of PSNR. However, the images estimated by MPRNet lack definition compared to the ground truth images. GibbsDDRM utilizes a generative model to generate components lost during the measurement process by considering the spectral space of the linear operator, which is one of the reasons why MPRNet outperforms GibbsDDRM in terms of PSNR. In addition, it is important to note that MPRNet is specifically trained on the corruption caused by motion blur.

In our experiments, other comparison methods, except for MPRNet, do not perform well in restoring the images with a high degree of accuracy. This is consistent with the results reported in (Chung et al., 2023a). In the motion blur corruption process used in this study, the blur kernel is relatively large to the image size, and there is also measurement noise, making it challenging to estimate a stable solution in such situations.

Relationship between hyperparameters η and η_b and each evaluation metric. We show the relationship between the hyperparameters η and η_b and each evaluation metric on the FFHQ dataset in Table 4. Note that although there is a small difference, the parameters that best achieve LPIPS differ from those that best achieve PSNR. The parameters for T, M_t, and Langevin dynamics are set to be the same as those described in the paper.

$\eta \setminus \eta_b$	0.7	0.8	0.9
0.7	40.57	39.43	38.70
0.8	40.59	39.28	**38.51**
0.9	42.00	40.68	39.50

(a) FID (↓)

$\eta \setminus \eta_b$	0.7	0.8	0.9
0.7	25.38	25.65	25.78
0.8	25.39	25.64	25.78
0.9	25.36	25.62	**25.81**

(b) PSNR(↑)

$\eta \setminus \eta_b$	0.7	0.8	0.9
0.7	0.125	0.118	**0.115**
0.8	0.125	0.119	**0.115**
0.9	0.130	0.123	0.118

(c) LPIPS(↓)

Table 4. Relationship between hyperparameters and evaluation metrics on FFHQ (256×256) dataset. **Bold**: Best.

Investigation of sampling methods of φ. In GibbsDDRM, φ is sampled by Langevin dynamics using the estimated score in (17). If no Gaussian noise is added in Eq. (11), the operation can be interpreted as a step of gradient descent method for maximum a posteriori (MAP) estimation of φ, with $\log p(\varphi|\mathbf{x}_{t:T}, \mathbf{y})$ as the likelihood function. Although this operation cannot be included in GibbsDDRM as it is not a sampling of φ, we can consider updating φ using this procedure. This strategy is referred to as "MAP" and the GibbsDDRM as "Langevin." Figure 7 shows histograms of PSNR and LPIPS computed for the images (in total 1000-images) estimated by Langevin (GibbsDDRM) and by MAP in the blind image deblurring experiment on FFHQ (256×256) dataset. It can be seen that the MAP's histogram has a longer tail, indicating that while MAP can sometimes estimate images with high accuracy, it is less stable compared to Langevin. This suggests that Langevin sampling serves to stabilize the estimation of φ.
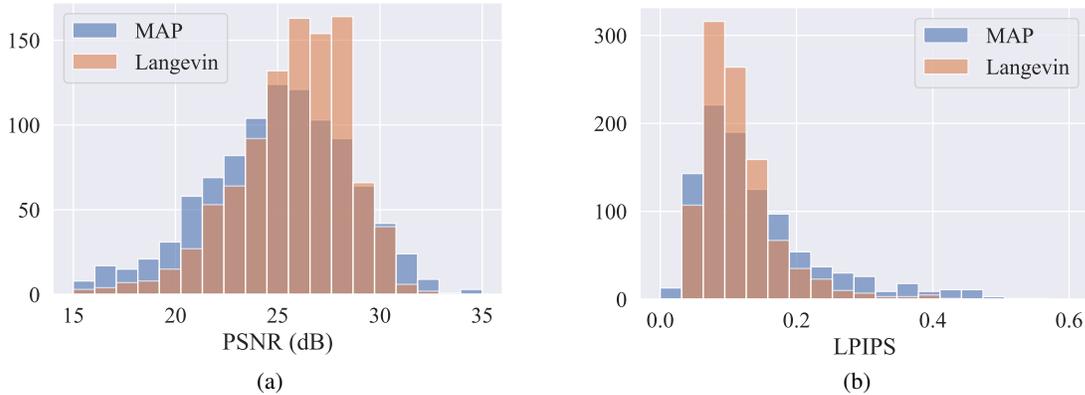

Figure 7. Histograms of blind image deblurring results on FFHQ (256×256) dataset obtained from different update strategies for φ. **MAP**: The linear operator's parameters are updated by MAP estimation, **Langevin**: GibbsDDRM, Proposed.

Additional figures We list additional qualitative results in Figs. 8 and 9 in order to see the details in the restored images and kernels.
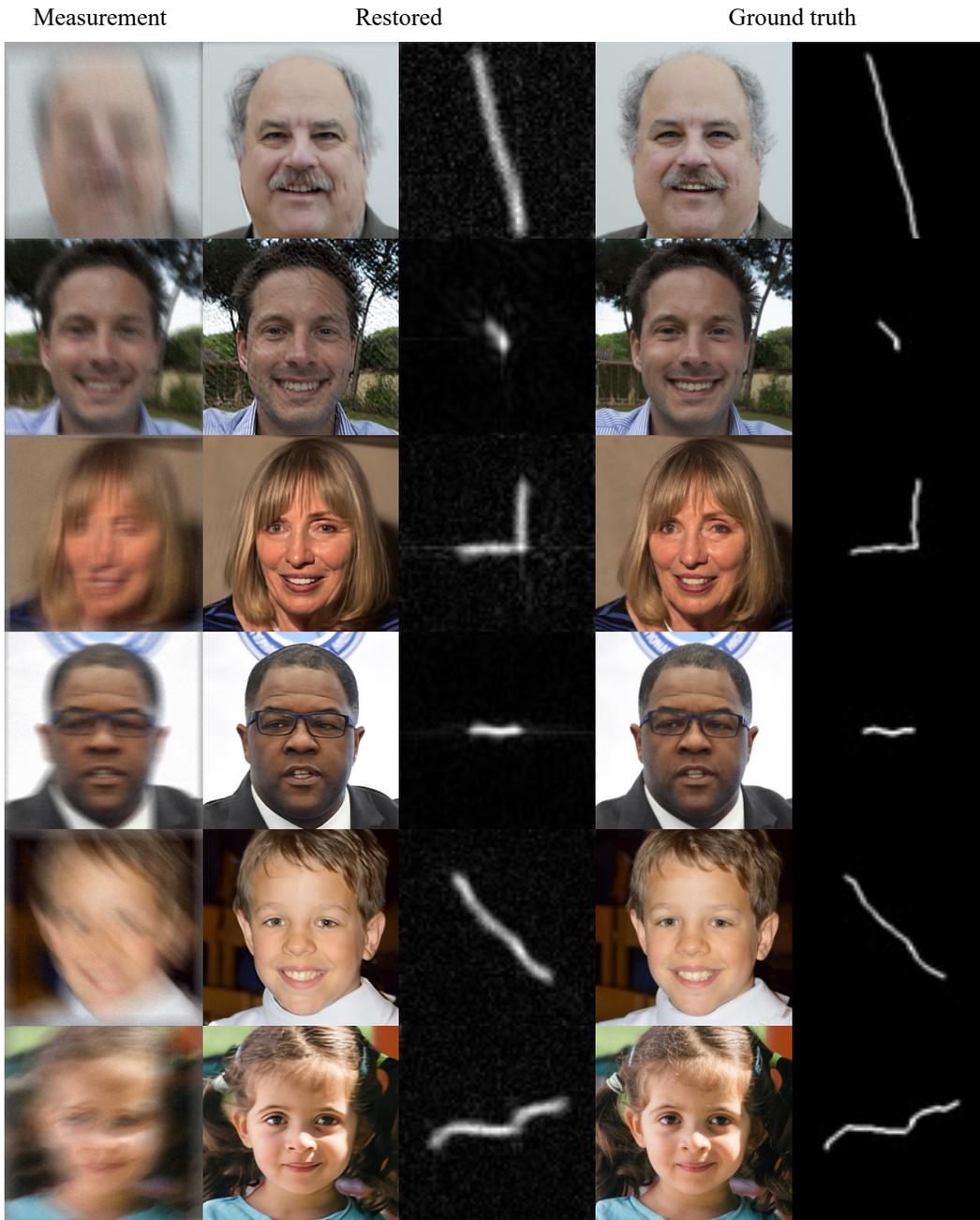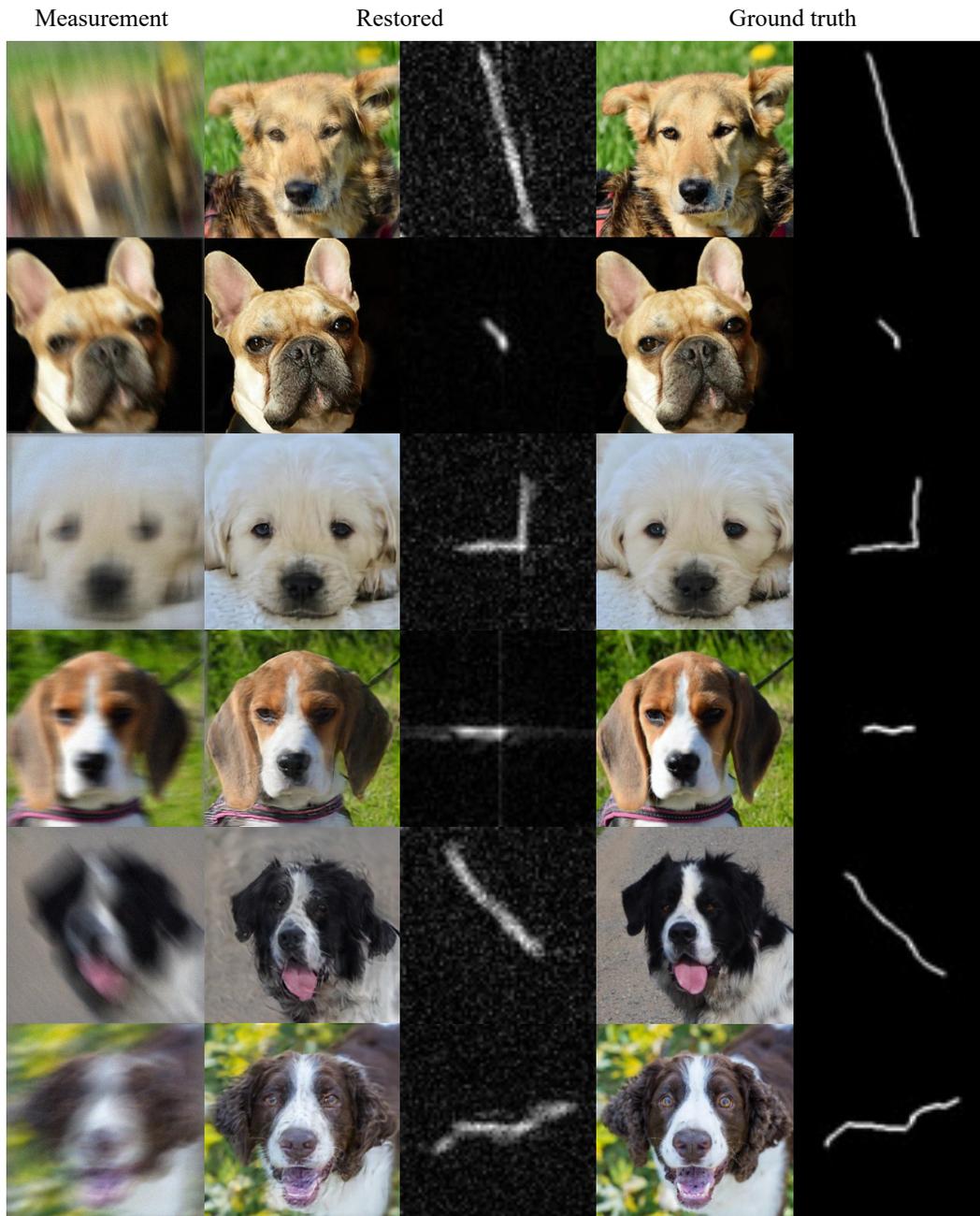
Measurement　　　　　Restored　　　　　Ground truth

Figure 8. Blind image deblurring results obtained by GibbsDDRM on FFHQ (256×256) dataset.

Measurement   Restored   Ground truth

Figure 9. Blind image deblurring results obtained by GibbsDDRM on AFHQ (256 × 256) dataset.